PROGRAMA DE BECAS EXCELENCIA SENESCYT EXTERIOR	
NOMBRE DEL BECARIO	Darwin Patricio Castillo Malla
UNIVERSIDAD	Universidad Politécnica de Madrid
TÍTULO OBTENIDO	Máster Universitario en Ingeniería Biomédica
TEMA DE TESIS	Desarrollo e Implementación de un Módulo de Videoconferencia a través de Webrtc para una Plataforma de Telemedicina Rural

UNIVERSIDAD POLITÉCNICA DE MADRID

ESCUELA TÉCNICA SUPERIOR DE
INGENIEROS DE TELECOMUNICACIÓN

TRABAJO FIN DE MÁSTER

MÁSTER UNIVERSITARIO EN INGENIERÍA BIOMÉDICA

DESARROLLO E IMPLEMENTACIÓN DE UN MÓDULO DE VIDEOCONFERENCIA A TRAVÉS DE WebRTC PARA UNA PLATAFORMA DE TELEMEDICINA RURAL

DARWIN PATRICIO CASTILLO MALLA

2014

UNIVERSIDAD POLITÉCNICA DE MADRID

ESCUELA TÉCNICA SUPERIOR DE
INGENIEROS DE TELECOMUNICACIÓN

Dpto. de Tecnología Fotónica

Grupo de Bioingeniería y Telemedicina

TRABAJO FIN DE MÁSTER

MÁSTER UNIVERSITARIO EN INGENIERÍA BIOMÉDICA

DESARROLLO E IMPLEMENTACIÓN DE UN MÓDULO DE VIDEOCONFERENCIA A TRAVÉS DE WebRTC PARA UNA PLATAFORMA DE TELEMEDICINA RURAL

DARWIN PATRICIO CASTILLO MALLA

2014

TRABAJO FIN DE MÁSTER

Título: **Desarrollo e implementación de un Módulo de Videoconferencia a través de WebRTC para una Plataforma de Telemedicina Rural.**

Autor: Castillo Malla Darwin Patricio

Tutor/a: Dª. María Elena Hernando Pérez

Tribunal:

Presidente:	D. Enrique J. Gómez Aguilera
Vocal:	D. Santiago Aguilera Navarro
Vocal secretario:	Dª. María Elena Hernando Perez
Suplente:	D. José Javier Serrano Olmedo

Fecha de lectura:

Calificación:

Resumen: La implicación de las TICs en la medicina, a día de hoy, ha traído consigo grandes beneficios a nivel sanitario, de investigación e incluso económico, de allí que la telemedicina juegue un papel importante en este proceso de continuo crecimiento. A través de la Telemedicina se permite brindar una continua asistencia y colaboración entre médico-paciente-médico rural, y aún más si es que esté paciente o médico se encuentra en lugares de difícil acceso. En el presente trabajo se plantea el desarrollo e implementación de un módulo de videoconferencia para una plataforma virtual de telemedicina rural diseñada por el proyecto de Telsalud UTPL Tutupaly y de la Uiversidad Técnica Particular de Loja, Ecuador en colaboración con el Grupo de Bioingeniería y Telemedicina (GBT) de la Universidad Politécnica de Madrid. La tecnología a utilizar es *open source*, concretamente WebRTC; generando de esta forma una aplicación sencilla y de fácil acceso al usuario únicamente a través de navegadores web.

Palabras clave: videoconferencia; videoconsulta; telemedicina; WebRTC; Sipml5; asterisk, webrtc2sip.

Title: Development and implementation of a module of videoconference with WebRTC for a Rural Telemedicine Platform

Summary: The involvement of ICT in medicine today, has brought great benefits to health status and even economic research, telemedicine hence play an important role in this process of continuous growth. Through Telemedicine is allowed providing continuous support and collaboration between doctor-patient-rural doctor, and even more if the patient or the doctor found in inaccessible places. The open source software is proposed for the development and implementation of a module of videoconference for a rural telemedicine platform. Specifically WebRTC technology is used and the results indicate a simple and easy way for the users because the application need only web browsers for its function.

Keywords: videoconference; WebRTC; telemedicine; sipml5; asterisk; webrtc2sip.

ÍNDICE

1. INTRODUCCIÓN 8

1.1 Presentación del Trabajo 8

1.2 Estructura del Trabajo 9

2. ANTECEDENTES 10

2.1 Breve descripción, telemedicina en la actualidad 10

2.2 Sistemas de Videoconferencia: 11

2.3 Entornos de Videoconferencia 15

3. JUSTIFICACIÓN Y OBJETIVOS 21

3.1 Justificación 21

3.2 Objetivos: 22

4. MATERIALES Y MÉTODOS 23

4.1 Identificación de Requisitos: 23

4.2 Esquema de Desarrollo de Módulo: 25

4.3 Herramientas Utilizadas: 28

5. RESULTADOS 30

5.1 Modelado: 30

5.2 Desarrollo e Implementación: 33

5.3 Desarrollo de interfaz de módulo de videoconferencia e integración con *webrtc2sip:* 38

5.4 Módulo de Videoconferencia en funcionamiento 40

6. CONCLUSIONES 46

7. TRABAJOS FUTUROS 47

8. BIBLIOGRAFÍA 48

9. ANEXOS 50

MANUAL DE USUARIO 50

Guía Técnica de Instalación de Webrtc2sip (V 2.5 -2013-10) 60

ÍNDICE DE IMÁGENES

Fig. 1 *Esquema de un equipo médico de videoconferencia [18].* *12*
Fig. 2 *Arquitectura de Webrtc2sip propuesta por Doubango Telecom para el establecimiento de comunicaciones [24].* *18*
Fig. 3 *Diagrama General de secuenciación de ingreso, registro y activación de sesión del modulo de Videoconferencia dentro de la Plataforma de Telemedicina Rural.* *26*
Fig. 4 *Diagrama de Actividades Generales del módulo de Videoconferencia para Teleconsulta Directa* . *27*
Fig. 5 *Arquitecturas de Webrtc2sip utilizada para el diseño del módulo de videoconferencia [24]*..... *28*
Fig. 6 *Diagrama de secuenciación y señalización de protocolos de transporte a utilizar en el módulo de videoconferencia [24].* *29*
Fig. 7 *Casos de Uso asociados al acceso del Módulo de Videoconferencia*..... *31*
Fig. 8 *Casos de Uso asociados a Gestionar Módulo* *32*
Fig. 9 *Casos de Uso asociados a usuarios principales del módulo: Médico Rural y Especialista.* *32*
Fig. 10 *Casos de Uso asociados a personal directivo/administrativo*..... *33*
Fig. 11*Diagrama Esquemático de la Plataforma General de Telemedicina Rural [27]* *34*
Fig. 12 *Interfaz Principal de ingreso a la Plataforma General de Telemedicina Rural [27].* *34*
Fig. 13 *Configuración de puertos a ser utilizados por la pasarela webrtc2sip.* *36*
Fig. 14 *Indicación de inicio de pasarela webrtc2sip y correcto funcionamiento de la misma.*..... *37*
Fig. 15 *Pruebas de ensayo de establecimiento de conexión de videoconferencia a través de WebRTC* *37*
Fig. 16 *Pestaña del módulo de Videoconsulta en la plataforma Telesalud UTPL Tutupaly*..... *38*
Fig. 17 *Interfaz del Módulo de Videoconferencia para médicos especialistas y médicos rurales, respectivamente.* *39*
Fig. 18 *Mensaje informativo al desactivar el módulo de Videoconferencia*..... *40*
Fig. 19 *Selección de médico especialista a comunicar en la interfaz de médico rural* *42*
Fig. 20 *Petición de acceso a cámara y micrófono de parte de quien realiza la video llamada* *42*
Fig. 21 *Petición de acceso a cámara y micrófono a quien se realiza la video llamada y además informe o aviso de la llamada entrante a través de sonidos.* *43*
Fig. 22 *Dados los permisos de acceso a cámara y micrófono, se da la posibilidad de responder o rechazar la videollamada.* *43*
Fig. 23 *Establecimiento de Videoconferencia entre usuario identificado como medico rural y especialista* *44*
Fig. 24 *Aspecto de Calendario de Médico Especialista.*..... *44*
Fig. 25 *Ingreso de Horario de disponibilidad de Calendario de Médico Especialista.* *45*
Fig. 26 *Aspecto de Calendario de Médico Rural para consulta de horario de médicos Especialistas.* *45*

ÍNDICE DE TABLAS

Tabla I.*Sistemas de Videoconferencia y funcionalidades*..... *18*
Tabla II. *Actores del sistema y sus correspondientes casos de uso* *30*

1. INTRODUCCIÓN

1.1 PRESENTACIÓN DEL TRABAJO

En la actualidad la telemedicina tiene especial relevancia en el ámbito de la prestación de servicios y atención óptima a personas que se encuentran en lugares remotos o entornos rurales alejados de los centros médicos especialistas.

El término telemedicina hace referencia a *"medicina a distancia"*, por tanto; de acuerdo con la Organización Mundial de la Salud (OMS), es entendida como la práctica del cuidado de la salud a través de la interacción de comunicaciones de audio, video y datos [1]. Contribuye sobremanera en lugares en que los servicios médicos son carentes o faltos de atención necesaria para su desempeño óptimo.

La telemedicina presta un servicio de relevancia y facilidad para el desarrollo de consultas, tratamientos, diagnósticos, y terapias a distancia, sin la necesidad de que el paciente tenga que viajar hasta el lugar de la consulta o especialista [2], y así también da la posibilidad de llegar a lugares remotos y accidentados geográficamente que muchas de las veces dificultan su ingreso, reflejando todo ello en costo y tiempo, entendiéndose esto último no solo en un entorno rural, sino también urbano.

Generalmente un servicio de telemedicina está compuesto por: teleconsulta, teleurgencias, videoconferencias, teleformación, historial clínico. En el caso de un sistema de telemedicina rural se debe contar con los componentes básicos de tele-consultorio: *médico tratante, médico especialista, y el medio tecnológico*; donde este último se refiere fundamentalmente a llamadas telefónicas, correo electrónico, videoconferencia, radioenlace, etc.

Un claro ejemplo de la aplicación de la telemedicina es la teledermatología, ya que de acuerdo con [3] en el mundo existe una alta prevalencia de consultas por afecciones cutáneas, que se contrapone a la escasa capacitación de los médicos de atención primaria en el manejo de estas enfermedades y a la desigual distribución y disponibilidad de dermatólogos. En 1995, Perednia y Allen introdujeron el vocablo *teledermatología* como el término que incorpora las tecnologías de la información y las telecomunicaciones para ofrecer servicios de dermatología a distancia [1, 4], luego en 1998, Ferrer Roca O. menciona al respecto: "en la teledermatología, el dermatólogo utiliza mecanismos de videoconferencia para ver al paciente en tiempo real, o puede recibir fotografías digitales en tiempo diferido" [5]. Debido a que la dermatología es una especialidad visual, recobran especial importancia las aplicaciones tecnológicas audiovisuales de datos: teleconsulta, y videoconferencia [1].

Desde la perspectiva médica del cuidado a un paciente, el sistema de consulta a través de videoconferencia es muy frecuentemente utilizado en el campo de la dermatología, cardiología, el cuidado de recuperación de heridas, neurología, la detección de drogas, tratamiento diabético y psiquiatría, como ejemplo de este último en [6] se manifiesta la utilización de la videoconferencia como facilidad para las emergencias de psiquiatría, este estudio indica una satisfacción por parte de los pacientes y una experiencia exitosa en lo que respecta a los análisis y diagnósticos obtenidos, con lo cual se da también crédito al hecho de que la telemedicina no solamente requiere de la utilización de la tecnología, sino también de la aceptación del paciente.

En el presente trabajo, vista la importancia de la videoconferencia en el campo de la telemedicina; se propone el investigar, desarrollar e implementar un módulo de videoconferencia para una plataforma de telemedicina rural. Para ello, se desplegará información detallada referentes a la búsqueda de tecnologías *open source* que permitan desarrollar el módulo y a la vez brinden la fiabilidad y posibilidad de establecer una comunicación multimedia de audio y video que contribuya en sobremanera a los usuarios (médicos rurales, médicos especialistas, pacientes) a tener un mejor diagnóstico sanitario. Este módulo a desarrollar, concretamente se enmarcará en la Plataforma de Telemedicina del proyecto Telesalud UTPL Tutupaly de la Universidad Técnica Particular de Loja, Ecuador en colaboración con el grupo de Bioingeniería y Telemedicina (GBT) de la Universidad Politécnica de Madrid.

1.2 ESTRUCTURA DEL TRABAJO

De acuerdo con las líneas de trabajo marcadas para el desarrollo del presente, se contempla la presentación de los antecedentes, en los que se describe brevemente la importancia de la videoconferencia en distintas aplicaciones médicas, así también se indicará el estado actual de la Plataforma de Telemedicina Rural del proyecto Telesalud UTPL Tutupaly, y de esta manera partir hacia una breve descripción de la solución tecnología propuesta para el desarrollo del módulo de videoconferencia.

Realizado lo anterior, se indica la correspondiente justificación y objetivos, de tal manera que desemboquen en el proceso de la identificación de requisitos, modelado de casos de uso, desarrollo de software, búsqueda y aplicación de herramientas, desarrollo e implementación de la funcionalidad del módulo de videoconferencia planteado para la plataforma de telemedicina rural del proyecto antes indicado.

Al finalizar el trabajo se adjuntan las respectivas conclusiones, recomendaciones; y la exposición de los trabajos y mejoras a desarrollar en un futuro. Adicional a esto se establece también la guía o manual de usuario final, junto a la guía técnica a seguir para el establecimiento de la videoconferencia, permitiendo con este último el realizar los ajustes y cambios necesarios para el buen funcionamiento del módulo.

2. ANTECEDENTES

Se conocen en general dos tipos de telemedicina: *síncrona* o también conocida como *telemedicina directa* la cual implica la presencia de los interlocutores. Para ello son necesarias dos unidades de videoconferencia (uno en cada sitio) y sus respectivas conexiones capaces de transmitir audio y video simultáneamente; y el otro tipo de telemedicina corresponde a la *asíncrona* o de *"almacenamiento y reenvío"*, implica la transferencia de imágenes y/o vídeo para su revisión médica e interpretación, no en tiempo real, sino en el momento en que el profesional tenga disponibilidad [1,7-10].

La actual plataforma de Telemedicina Rural del proyecto de Telemedicina Rural "Telesalud UTPL Tutupaly", base de este trabajo, es de tipo *asíncrono*, y he allí la necesidad de desarrollar un módulo de videoconferencia que permita una *Teleconsulta Directa o Síncrona* que sea complementaria a esta plataforma.

Se propone por tanto, el desarrollo del módulo de videoconferencia a través de la utilización de software *open source*, por lo que en los siguientes párrafos se realizará una breve descripción de los diversos ítems que componen un sistema de videoconferencia y también se indicarán algunas de las conceptos y definiciones de los que el presente proyecto se servirá para acometer su objetivo fin.

2.1 BREVE DESCRIPCIÓN, TELEMEDICINA EN LA ACTUALIDAD

La telemedicina actualmente y dependiendo del lugar, país y situación constituye una aplicación muy diversa, siendo que, por lo general en Europa, Japón, Estados Unidos, Australia, se cuenta con una gran cantidad y diversidad de proyectos e iniciativas de un pronto desarrollo y aplicación.

Ejemplo: en España, el grupo de Bioingeniería y Telemedicina de la ETSIT de la UPM junto a otras instituciones pusieron en marcha el proyecto *"Sistema de Teleasistencia en Centros Escolares"*, el cual consistía en el establecimiento de televisitas entre un colegio/guardería y un hospital, apoyada sobre una videoconferencia y en un entorno web, para de esta manera resolver situaciones de asistencia médica no urgente [12].

En Estados Unidos el desarrollo de la telemedicina inició principalmente en entornos que presentan situaciones de catástrofes, un ejemplo constituye el mantenimiento de un programa que utiliza dos vías de audio interactivo y transmisiones de vídeo de movimiento completo unidireccionales de Armenia a los Estados Unidos, luego en la actualidad se encuentra muy desarrollada en los entornos urbanos y rurales [5, 13].

En países de América Latina, la situación es distinta dependiendo del país y lugar al que requiera la aplicación: Brasil lleva varios programas de asistencia de telemedicina en los lugares selváticos [14]. En Ecuador, actualmente se cuenta con el proyecto Tutupaly y con otros proyectos e iniciativas pequeñas de Telesalud en las principales ciudades [15].

En determinados países de África, la situación de telemedicina rural en general es muy limitada debido a las pocas posibilidades de tecnología, transporte y comunicaciones presentes. En 2001 la Universidad y escuela Médica de Mali en Bamako, junto a la financiación del gobierno de Ginebra, iniciaron el proyecto RAFT (*Réseau Afrique Francophone de Télémédiciné)* con el objetivo de establecer un sistema de telemedicina de bajo costo, de acuerdo con las necesidades locales y un desarrollo sustentable [17].

2.2 SISTEMAS DE VIDEOCONFERENCIA:

Los sistemas de videoconferencia en medicina toman importancia y utilidad en multitud de aplicaciones: educación médica personalizada, teleconsulta, educación al paciente, y atención directa al paciente [18].

2.1.1 Videoconferencia o Tiempo Real

La video consulta conlleva el utilizar equipamiento de videoconferencia que establezca una comunicación entre el paciente y su médico tratante, el cual muy a menudo se encuentra muy distante. En [6,7, 19] se manifiesta que la efectividad de estas consultas se encuentra entre el 67% y 80% en comparación con las consultas tradicionales; entendiéndose como consultas tradicionales, la consulta en la que el paciente se encuentra en un mismo sitio frente al especialista.

Los sistemas de videoconferencia en los últimos años han tenido un crecimiento importante debido a que sus componentes tecnológicos se han mejorado, esto es: incremento de ancho de banda, resolución de cámaras, monitores, codificadores, y decodificadores (CODECs) [1]. En la figura 1 se indica un esquema de un equipo de videoconferencia médica tradicional.

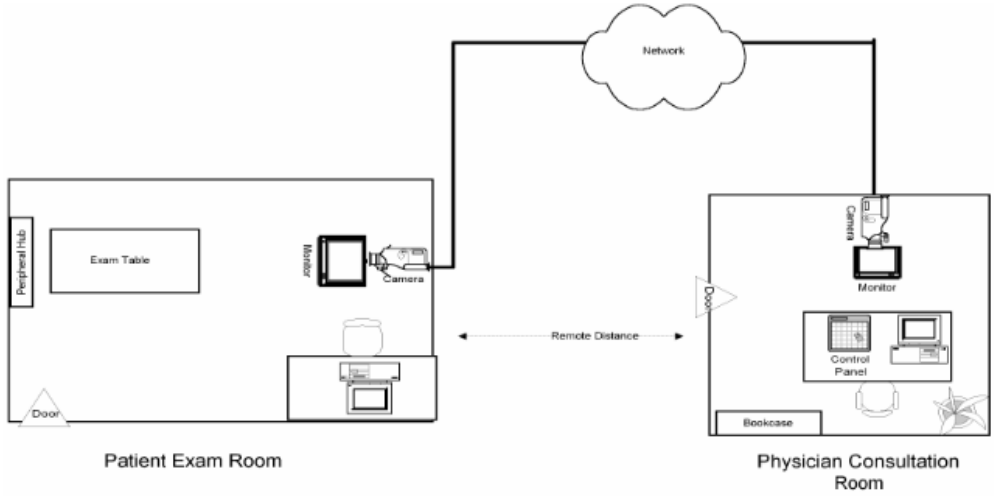


Fig. 1 Esquema de un equipo médico de videoconferencia [18].

De acuerdo con [18] para lograr un buen desempeño de un sistema de videoconferencia médico se tiene que tener en cuenta y cumplir ciertos roles, que involucran un equipo interdisciplinario de personas. Estos son:

- ✓ *Administrador de Telemedicina:* será el responsable del diseño, políticas y procedimientos de telemedicina.
- ✓ *Investigador de Telemedicina:* es el responsable de la recolección y análisis de los datos de la utilización del sistema.
- ✓ *Vendedor de Videoconferencia, especializado en Telemedicina:* responsable de los aspectos técnicos de selección, soporte, implementación y comunicación del equipamiento.
- ✓ *Técnico de Videoconferencia:* ingeniero responsable de la operación y mantenimiento del sistema de telemedicina.
- ✓ *Educador de Telemedicina:* responsable de la enseñanza y entrenamiento de los usuarios de telemedicina y sus asociados.
- ✓ *Usuarios de Telemedicina:* engloba al personal médico (médicos, especialistas, enfermeros/as) responsables de la atención directa del paciente y de la orientación del paciente en el involucramiento de la videoconferencia médica.

2.1.2 Características de Calidad de un sistema de Videoconferencia

En un equipo médico de videoconferencia, el contar con adecuados atributos de calidad, es de suma importancia, ya que estos son el reflejo de los requerimientos y criterios subsecuentes para el diseño, implementación y evaluación de actividades. Los modelos de atributos de calidad permiten y facilitan a los investigadores proveer una perspectiva, una terminología y un entendimiento común en la calidad de todas las variables envueltas en el fenómeno de la telemedicina [18].

De acuerdo con algunos autores, para tener un buen sistema de calidad desde un punto de vista práctico y utilitario, se debe "Empezar con el usuario, es decir determinar la situación clínica y de esta forma seleccionar la tecnología adecuada y no al contrario, ya que lo último es lo que conlleva a que existan fracasos en los proyectos planteados" [18].

Se tiene de esta manera que un equipo de videoconferencia para ser utilizado en medicina tiene que cumplir con ciertas características de tecnología, usabilidad, entorno y elementos humanos.

Atributos de tecnología: En este se encuentran todos los parámetros relacionados con el equipamiento y su proceso de comunicación en un entorno médico [18].

- ✓ *Manipulación del movimiento*: El tener grandes anchos de banda y poderosos CODECs permiten la reducción de la pixelación, los artefactos de movimiento, la caída de frames, los cuales afectan negativamente la calidad del video. La importancia de la manipulación del movimiento radica en cuando se tienen rápidos movimientos, tales como el angiograma de un corazón adulto o el latir del corazón de un feto.
- ✓ *Resolución de Imagen:* Se refiere a la representación de la imagen y su mejor fidelidad posible. En la videoconferencia de un sistema de telemedicina, la calidad de la resolución viene determinada por la resolución de la cámara que está recibiendo la imagen y el monitor que se encuentra proyectando dicha imagen. En este apartado intervienen también parámetros de utilidad como el zoom y su calidad, esto específicamente es muy utilizado en dermatología para indicar las lesiones de piel.
- ✓ *Claridad de audio:* Los niveles de audio deberían ser lo más cercano a los naturales tanto como sea posible [19, 20], de esta forma se evitaría la pérdida de datos y la confusión con ruido añadido.
- ✓ *Sincronización:* Considerado como el más alto estándar de calidad ya que la sincronización en audio y video en tiempo real, evita los retrasos y distracción de los pacientes.
- ✓ *Confiabilidad:* Un sistema y equipo de telemedicina debe brindar la confiabilidad suficiente, ya que de otra forma no se logrará una aceptación de la utilización del sistema por parte de los usuarios.
- ✓ *Sofisticación de periféricos:* Esto es referido a tener la suficiente tecnología que permita la extensión y adaptación del equipamiento médico de videoconferencia con otros dispositivos, tales como estetoscopio electrónico, cámara de dermatología, cámara de retina, electrocardiograma electrónico, etc.
- ✓ *Ergonomía:* el equipamiento debe ser diseñado de tal forma que permita una eficiente movilidad y facilidad de transporte.

- ✓ *Interoperabilidad:* Debe prestar la suficiente apertura para cualquier tipo de comunicaciones de tal forma que trabajen como un conjunto.

Atributos de usabilidad: Se refieren principalmente a la capacidad operacional de los equipos y dispositivos, ya que de esta forma se tendrá un impacto mayor de aceptación. Se pueden resumir en la siguiente lista:

- ❖ *Facilidad de uso:* El tener una interfaz agradable facilita la operación del equipo del usuario final.
- ❖ *Facilidad de Aprendizaje/Entrenamiento:* Corresponde a la claridad y entendimiento de las funciones en la operación de todos los actores implicados en el sistema.
- ❖ *Utilidad:* Indica el grado de funcionamiento y contribución al cuidado del paciente en un entorno clínico.
- ❖ *Accesibilidad y Disponibilidad, Seguridad, Atención centrada en el paciente.*

Atributos del entorno físico: referido a las condiciones médicas en las que se desarrollan el entorno de videoconferencia, los consultorios físicos como tales, su decoración, amueblamiento, espacio, privacidad y todo lo concerniente a la adaptación y confort con el equipamiento y personal.

Elementos Humanos: va en relación en cuanto a la adaptabilidad, la relación entre paciente-médico y equipamiento, el tener un personal de soporte, orientación y coordinación.

2.1.3 Ancho de Banda y conexión

El ancho de banda en los sistemas de videoconferencia es muy importante en aplicaciones médicas debido al cuidado de las situaciones que se desarrollan, como es el caso de telecirugía, teledermatología, monitorización cardiaca, ultrasonido y en los análisis de tiempo real.

Un centro de telemedicina debe contar con un suficiente ancho de banda que permita el establecimiento de la conexión sin retrasos, y posibles caídas del sistema, que garantice la suficiente velocidad en la transmisión de datos. A ser posible se debe tener una conexión Wi-Fi o Wi-Max para tener una mejor calidad y una alta velocidad en la transmisión de los datos [1].

La combinación de telemedicina síncrona y asíncrona son en este sentido de muy buena aplicabilidad, en el caso de que existiere pérdida de información en alguno de los terminales.

A nivel general, según la clasificación establecida en [21,22], en relación con los tipos de videoconferencia y su conexión de ancho de banda, se tiene:

- ✓ ***Videoconferencia personal de baja calidad***: ideal para conversaciones informales de 2 personas. Transmite en un rango de velocidades entre los 28.8 y los 64 Kbps sobre líneas telefónicas.
- ✓ ***Videoconferencia de escritorio***: usado por grupos pequeños de individuos, por ejemplo, reuniones hasta de 4 personas. Opera en el rango de velocidades entre los 64 y 128 Kbps.
- ✓ ***Videoconferencia de calidad intermedia:*** ideal para reuniones en torno a una mesa (hasta 15 personas). Se transmite en un rango de velocidades entre los 128 y los 384 Kbps. Sobre tecnología IP en internet.
- ✓ ***Videoconferencia de alta calidad:*** Necesaria para grandes reuniones. Opera en un rango de velocidad entre 384 Kbps y los 2 Mbps.

2.3 ENTORNOS DE VIDEOCONFERENCIA

En esta sección se detallarán algunas de los sistemas y entornos de videoconferencia que se utilizan actualmente y junto a ellos sus respectivos protocolos de comunicación.

Windows live Messenger: Es un cliente de mensajería instantánea creada por Microsoft, diseñado actualmente para funcionar en PC con Microsoft Windows, móviles con Windows phone, e iOs. Se caracteriza principalmente por tener:

- ✓ Llamadas de PC a teléfono
- ✓ Interoperabilidad con Yahoo Messenger que permite a los usuarios de ambas charlar entre sí sin necesidad de crear una cuenta en el otro servicio.
- ✓ Mensajes sin conexión.
- ✓ Videojuegos y aplicaciones compartidas a través de la ventana de conversación.
- ✓ Carpetas compartidas como alternativa para el método de transferencia directa de archivos en las ventanas de conversación. Windows Live SkyDrive.

Skype: Se considera uno de los más populares software que permite comunicaciones de texto, voz y vídeo sobre internet (VoIP). Fue diseñado en 2003 en Tallin, Estonia, por el danés Janus Friis y el sueco Niklas Zennström y desarrollada en su solución técnica por los estonios Priit Kasesalu, y Ahti Heinla. El código y protocolo de Skype permanecen cerrados y son exclusivos de sus propietarios. Los usuarios de Skype pueden hablar entre ellos gratuitamente, a través de voz y video. Ofrece llamadas de bajo coste, entre ordenador y teléfono móvil. Conferencia grupal de pago.

VaaS: Servicio de videoconferencia desarrollado por el Departamento de Ingeniería Telemática (DIT*)* de la ETSIT de la Universidad Politécnica de Madrid. Se

trata de un servicio multipunto en la nube basado en *Isabelv5 (aplicación de videoconferencia)* que se apoya en un servidor que otorga acceso a clientes tanto desde PC fijo/portátil, como desde terminales móviles.

El servidor de multivideoconferencia es capaz de interconectar PCs con terminales móviles, a través de navegador web y sin necesidad de instalar aplicaciones. Sus principales características son:

- ✓ El servidor se realiza adaptando Isabelv5, el Gateway flash y otras tecnologías asociadas para soportar los siguientes terminales en la videoconferencia:

  a. PC con navegador con soporte de video Flash
  b. Terminal móvil capaz de emitir y recibir video en HTML5
  c. Terminal móvil tipo iPad capaz de emitir y recibir video en HTML5.

- ✓ Capacidad para soportar varias multiconferencias simultáneas e independientes, dependiendo de las características del servidor.
- ✓ El sistema incluye una gestión mínima de multiconferencias.
- ✓ Gestión de estadística de códecs, uso de CPU, uso de BW, resolución, etc.

Tango Video Calls: Aplicación de videoconferencia disponible para PC, iOS, Android y Windows Phone. Permite interoperabilidad entre las diferentes plataformas y realizar llamadas de forma gratuita sobre Wifi y 3G. Una de las primeras aplicaciones para dispositivos móviles.

Facetime: Aplicación nativa y de videoconferencia propiedad de Apple para sus dispositivos iPhone, iPad, Mac y iPod touch. Únicamente con redes wifi, excluye 3G y permite transmitir vídeo capturado tanto con la cámara frontal como la posterior.

Google Hangouts: Servicio de multiconferencia de google que opera en web. Las aplicaciones móviles no soportan multiconferencia. Soporta hasta 10 participantes en la multiconferencia.

Big Blue Button: Aplicación web open source para videoconferencia y e-learning. Distribuido bajo licencia GNU. Es producto de la reutilización de proyectos como Asterisk, Flex SDK, Red5, MySQL, y otros.Ofrece la posibilidad de que sus usuarios suban archivos PDF, docs, xls. Entre sus características se encuentran:

- ✓ *Presentador,* puede subir presentaciones y compartir su escritorio.
- ✓ *Espectador,* no tiene autoridad en la videoconferencia, solo puede ver o chatear.
- ✓ *Moderador,* puede subir presentaciones, compartir su escritorio y aceptar o expulsar usuarios.

- ✓ El funcionamiento del video está formado por una serie de frames (imágenes fijas) que irán presentándose en pantalla a una velocidad determinada dando lugar al video. De esta forma cuantos más frames, más ágil la reproducción de video. El problema del envío de estos frames a través de la red es que normalmente no pueden enviarse en un solo paquete debido a su tamaño. Por lo tanto, cada frame llegará al dispositivo dividido en varios paquetes y la labor del software es reconstruir cada frame para presentarlo.

Proyecto WebRTC

WebRTC, cuyas siglas hacen referencia a (Web Real-Time Communication) es un proyecto libre y abierto (*open source*), apoyado por Google, Mozilla y Opera; que consiste en el desarrollo de una API (interfaz de programación de aplicaciones) que permite a los navegadores web el establecimiento de comunicación multimedia en tiempo real (RTC); mediante desarrollo de programación en JavaScript y HTML5 [23], de esta forma su principal objetivo es la *no utilización de plugins adicionales*. El desarrollo de es promovido por el equipo de Google Chrome.

- ❖ ***HTML5:*** Corresponde a la quinta revisión del lenguaje HTML, permite la creación de elementos dinámicos y la definición de WebSockets como protocolos de comunicación entre navegadores y servidores. Es promovido principalmente por el consorcio W3C *(World Wide Web Consortium)*.

- ❖ ***SIPML5:*** Es una plataforma de cliente HTML5 SIP del lado del cliente completamente abierta y de licencia open source, escrito en JavaScript para la integración en las redes sociales, juegos en línea, sitios web de comercio electrónico, firmas de correo electrónico. Cimienta su comunicación en WebRTC, ha sido desarrollado por la organización Doubango. Esta plataforma permite al usuario utilizar como protocolo de conexión cualquier red IMS SIP o a su vez, simplemente utilizar los servicios ofertados por la organización para la conexión [24].

Webrtc2sip

Es un software de lado del servidor de libre distribución bajo licencia GNU, desarrollado por Doubango Telecom[24]. Constituye una pasarela inteligente que permite la conexión y establecimiento de comunicación multimedia entre terminales que soporten WebRTC, utilizando para ello el protocolo de comunicaciones SIP (*Session Initiation Protocol*).

En la siguiente figura se indica la arquitectura propuesta por Doubango, la cual la componen cuatro módulos: *Proxy SIP, WebRTC Breaker, Codificador Multimedia y servicio de click-to-call.*

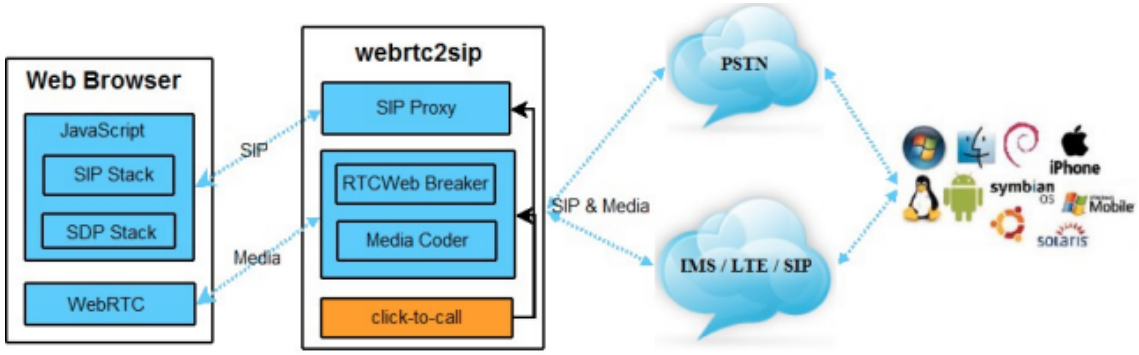


Fig. 2 Arquitectura de Webrtc2sip propuesta por Doubango Telecom para el establecimiento de comunicaciones [24].

A continuación se presenta una tabla que resume alguno de los sistemas ya enunciados con sus principales funcionalidades:

Tabla I. Sistemas de Videoconferencia y funcionalidades.

Sistema Videoconferencia	**Arquitectura y Requerimientos**	**Funcionalidades**
Skype	Aplicación móvil y Web. Requiere instalar aplicación Utiliza protocolo SIP.	*Mensajes de video. *Envío de archivos. *Alternar entre chat. *Videollamada grupal (de pago). *Gestión de Agendas *Compartir documentos y pantalla. *registro de llamadas, datos y mensajes, temporalmente.
Open Meeting	Instalar aplicación Web Open Source Flash Player Java Runtime Environment	*Videoconferencia *Mensajería Instantánea. *Compartir documentos y pantalla. *Organización a través de salas de conferencia (rooms), participantes entre 2 y 1000. * Agrupar participantes.
Ovoo	Web Instalar aplicación	*Aplicación gratuita y de pago. *Videoconferencia compartida hasta 12 personas. *Mensajería Instantánea.
WebRTC	Aplicación Web *HTML5	*Funcionalidad para adecuar a las necesidades requeridas del presente

	*JavaScript APIs Funciona directamente en navegador Chrome, Firefox, Opera	proyecto.

Protocolos de Comunicación :

IP Multimedia Subsystem; O también llamado subsistema de red multimedia es una arquitectura desarrollada para los servicios multimedia IP, fue inicialmente desarrollada por 3GPP *(3rd Generation Partnership Project)* como parte del desarrollo de las redes móviles 3G.

Para facilitar la integración con Internet, IMS usa protocolos IETF *(Internet Engineering Task Force),* por ejemplo SIP *(Session Initial Protocol)*. De acuerdo con 3GPP, IMS no está destinada a estandarizar aplicaciones sino más bien para ayudar al acceso de las aplicaciones multimedia y de voz desde terminales inalámbricos y de línea fija, es decir, para crear una forma de convergencia fijo-móvil (FMC).

SIP: Es un protocolo de internet diseñado para establecer sesiones entre usuarios. Puede ser utilizado para Telefonía IP (voz y video), presencia, mensajería instantánea, conferencias y más [25]. Es un protocolo desarrollado por MMUSIC del IETF (RFC 3261) con la intención de ser el estándar para la iniciación, modificación y finalización de sesiones interactivas de usuario donde intervienen elementos multimedia como el video, voz, mensajería instantánea, juegos en línea y realidad virtual.

La sintaxis de sus operaciones se asemeja a las de HTTP y SMTP, que son protocolos utilizados en páginas web y distribución de emails, respectivamente; esto es por cuanto SIP fue diseñado para ser un servicio más de internet. En noviembre de 2000, SIP fue aceptado como el protocolo de señalización de 3GPP y elemento permanente de arquitectura IMS.

SIP es un protocolo a nivel de aplicación desarrollado por el IETF, se utiliza para establecer, conducir y terminar sesiones multimedia y multiusuario sobre redes TCP/IP usando cualquier tipo de medio. Habitualmente es utilizado para VoIP, pero podría ser utilizado también para establecer comunicaciones de tipo *streamming* entre "end points".

SIP emplea un modelo de transacciones petición-respuesta similar a HTTP para las comunicaciones entre clientes finales. Utiliza generalmente los puertos 5060 y/ó 5061 (tanto TCP como UDP). El puerto 5060 se usa para las sesiones de tráfico de señalización de SIP no cifrado, y el puerto 5061 suele ser utilizado para las sesiones SIP cifradas mediante TLS (transport Layer Security).Permite interoperabilidad, escalabilidad e interconexión globales.

Arquitectura SIP: SIP soporta funcionalidades para el establecimiento y finalización de las sesiones multimedia: localización, disponibilidad, utilización de recursos, y características de negociación. Existen dos elementos de red fundamentales, los agentes de usuario (UA) y los servidores.

1. User Agents (UA): terminales de usuario. Compuestos por dos elementos:
   1.1 User Agent Client (UAC), entidad lógica que genera peticiones SIP y recibe respuestas a esas peticiones.
   1.2 User Agent Server (UAS); entidad lógica que genera respuestas a las peticiones SIP. Es decir, procesa y responde a las peticiones de UACs.

Estos se encuentran en todos los agentes de usuario, de esta manera permiten la comunicación cliente-servidor.

2. Los servidores SIP, tres tipos:

   2.1 Servidor Proxy: se ocupa de retransmitir solicitudes y decidir a que otro servidor deben remitir, alterando los campos de la solicitud en caso de ser necesario. Actúa como cliente y servidor para establecer llamadas entre usuarios. Funcionalidad semejante a un servidor Proxy HTTP que tiene la tarea de encaminar las peticiones que recibe a otras entidades más próximas al destinatario.
   2.2 Servidores de Registro: permite establecer la ubicación física de un usuario determinado, esto es, en qué punto de la red está conectado.
   2.3 Servidor de Localización: simplemente da información acerca de donde puede estar el cliente al que se quiere llamar para así poder localizarlo.

Asterisk:

Asterisk es un software PBX diseñado con el concepto de software libre (GPL). por la empresa Digium, misma que se centra en el desenvolvimiento de código fuente y en hardware de telefonía de bajo costo que funciona con Asterisk.

El software Asterisk corre en plataforma Linux y otras plataformas Unix con o sin hardware. Permite la unificación y conectividad de tecnologías: VoIP, GSM y PSTN. [26].

Reconoce muchos protocolos VoIP como pueden ser SIP, H.323, IAX y MGCP. Asterisk puede interoperar con terminales IP actuando como un *registrador* y como *gateway* entre ambos [26].

3. JUSTIFICACIÓN Y OBJETIVOS

3.1 JUSTIFICACIÓN

Dada la importancia e implicación que actualmente tienen las tecnologías de información y comunicación en el campo de la medicina, como medio de ayuda, desarrollo y contribución a la mejora de la calidad de asistencia sanitaria de pacientes y la formación de médicos, especialmente los que se encuentran en lugares aislados o de difícil acceso, surge la necesidad de buscar alternativas y soluciones que requieran de un bajo coste y eficiencia.

En este sentido, el proyecto de Telemedicina Rural *"Telesalud UTPL Tutupaly"* de la Universidad Técnica Particular de Loja de Ecuador en colaboración con el grupo de Bioingeniería y Telemedicina de la Universidad Politécnica de Madrid [27], actualmente conservan una plataforma virtual de telemedicina rural de tipo asíncrono desarrollada con tecnología *open source*; por lo que un paso más en el caminar de este proyecto es el brindar servicios de Telemedicina Directa en tiempo real, y es allí precisamente donde nace la necesidad de llevar a cabo el desarrollo del presente trabajo con la creación de un módulo de videoconferencia que contribuya a la consecución de este objetivo último de la plataforma.

Ciertamente que en el momento presente se cuenta con diversidad de software que permiten establecer videoconferencia, no obstante, muchos de ellos y en gran medida dependiendo de su calidad son servicios con costes adicionales. Por tanto, en el presente proyecto se propone la investigación y búsqueda de información y procedimientos que permitan realizar este módulo de videoconferencia a través de software open source.

Se ha identificado como alternativa de solución y aplicabilidad para el desarrollo de este módulo, el empleo de *WebRTC, Html5, Asterisk y Webrtc2sip*; todos estos últimos son de libre distribución y por tanto factibles de adaptar a los componentes presentes de la plataforma general.

Por indicar una de las ventajas principales de la utilización de WebRTC, es que no se necesitaría la instalación de plugins o paquetes adicionales en el momento de la utilización del módulo, sino únicamente el contar con una conexión a internet y navegadores Mozilla Firefox o Google Chrome; lo cual facilitaría al usuario la utilización y le evitaría la instalación de plugins o paquetes adicionales, lo que muchas de las veces hace que el usuario sea temeroso y a su vez genere un rechazo a la aplicación.

3.2 OBJETIVOS:

El objetivo general del presente trabajo constituye el desarrollo e implementación de un módulo de videoconferencia basado en tecnología *open source* para la Plataforma de Telemedicina Rural del proyecto Telesalud Tutupaly de la Universidad Técnica Particular de Loja de Ecuador en colaboración con la Universidad Politécnica de Madrid.

Para el cumplimiento del mismo se contemplan los siguientes objetivos específicos:

- ✓ Recopilar información referente a herramientas tecnológicas y software *open source* que permitan el desarrollo del módulo de videocoferencia.

- ✓ Desarrollar el módulo de videoconferencia para realizar teleconsultas en directo mediante la aplicación directa de la tecnología WebRTC.

- ✓ Implementar el módulo de videoconferencia con tecnología *open source* y adaptarlo en la plataforma de Telemedicina Rural Telesalud UTPL Tutupaly.

4. MATERIALES Y MÉTODOS

En la presente sección se desarrollaran e indicaran detalles referentes a los componentes tecnológicos utilizados para el desarrollo e implementación del módulo de Videoconferencia, así también se abordará la identificación de requisitos y requerimientos específicos del sistema.

4.1 IDENTIFICACIÓN DE REQUISITOS:

4.1.1 Actores:

En conjunto con la identificación adecuada de los requerimientos del sistema se ha considerado los siguientes aspectos y actores principales en la utilización del módulo a desarrollar:

Convergencia de Tecnologías: Se refiere al aspecto de combinar adecuadamente los aspectos de la comunicación en tiempo real a través de la tecnología *WebRTC*, para así obtener un módulo sencillo, innovador, capaz de soportar la comunicación de audio y video.

Usuarios: Se ha identificado tres tipos de usuarios principales de la aplicación, estos son:

- ***Médico rural:*** Es la persona que se encuentra en un entorno remoto, rural, en términos sencillos es quien iniciará primordialmente una videoconsulta de acuerdo con las necesidades avenidas a el mismo, sean las mismas de atención de especialidades concretas o consultas a nivel general sanitario o de educación.
- ***Médico Especialista:*** Fundamentalmente es la persona que estará disponible para las consultas en directo o video-consultas de acuerdo con un horario establecido y organizado de acuerdo con su disponibilidad.
- ***Personal Administrativo y/o Directivo:*** Son las personas que podrán tener tanto el rol de médico rural como de especialista de tal forma que pueda interactuar con cualquier grupo indistintamente.
- ***Administrador del Sistema:*** Es la persona que se encarga de administrar y manejar los datos de los usuarios a nivel de sistema, será a su vez quien esté pendiente del funcionamiento adecuado del módulo, así como también de la gestión de cambios y adecuaciones del sistema desarrollado dentro de la plataforma general de telemedicina.

4.1.2 Requisitos Funcionales:

RF-001: Ingreso de usuarios: El sistema deberá permitir la comprobación del ingreso de usuarios al sistema principal de la plataforma y además a la sección

correspondiente a la videoconferencia. El dar de alta a los usuarios, sean estos médicos rurales, especialistas, directivos, etc., está fuera del alcance de este trabajo, ya que depende exclusivamente del ingreso principal al sistema.

RF-002: Modificar datos de usuarios: La modificación de datos de usuario, respecto a sus correspondientes roles, estará dado directamente como una función del administrador del sistema en general.

RF-003: Baja usuario: Será una función directa del administrador del sistema principal, el cual dará de baja a cualquier usuario que así lo solicite el personal directivo o administrativo.

RF-004: Activación de Videoconferencia: Los usuarios identificados tanto como médicos rurales, especialistas o directivos tendrán la posibilidad de establecer la conexión y disponibilidad de presencia al sistema de videoconferencia. Para ello bastará con activar la funcionalidad destinada a efecto del mismo.

RF-004: Identificación de presencia: El módulo propuesto permitirá identificar a los usuarios que se encuentren conectados al sistema de videoconferencia, indicando su disponibilidad o su horario correspondiente en caso de que no está disponible.

RF-005: Consultar horario de Disponibilidad: Todos los usuarios del sistema podrán consultar la disponibilidad de los usuarios identificados como médicos rurales, especialistas y personal directivo.

RF-006: Registrar Horario de Disponibilidad: Será una función específica de los médicos especialistas, los cuales podrán registrar en el calendario su hora y día correspondientes de acuerdo con su posibilidad de estar disponible para atender las video consultas. El sistema almacenará esta información asociada al médico especialista que la ha registrado.

RF-007: Salida del Sistema de Videoconferencia: Corresponderá a la salida única del sistema de videoconferencia, lo cual no impedirá al usuario seguir moviéndose por el resto de funcionalidades de la plataforma web.

RF-008: Consultar Manual de Usuario: Todos los usuarios del sistema independientemente de su rol, tendrán lo posibilidad de consultar el manual de usuario del módulo de videoconferencia.

4.1.3 Requisitos de Rendimiento:

RD-001: Tiempo de procesamiento y actualización: El tiempo de procesamiento y de actualización del estado del usuario, será el determinado por la red de comunicaciones, La información registrada y modificada en el sistema como horarios y

estado del usuario (disponible o no disponible) se actualizará en tiempo real, indicando ello al resto de usuarios del sistema.

RD-002: Conexión de usuarios: El módulo de videoconferencia será capaz de soportar diferentes sesiones de videoconferencia al mismo tiempo.

4.1.4 Requisitos de Información:

RI-001: Datos de usuarios: La información de los usuarios que el módulo de videoconferencia maneje será: identificación de rol de usuario (médico rural, médico especialista, directivo), y su estado de disponibilidad.

4.1.5 Requisitos de Operación

RO-001: Inicio de Sesión: Para que el usuario utilice el módulo de videoconferencia, tendrá que acceder a la plataforma de Telemedicina y luego dirigirse a la sección de videoconferencia destinada para tal efecto dentro de la plataforma.

RO-002: Establecimiento de Conexión: La videoconferencia se establecerá automáticamente cuando el usuario haya sido registrado en el sistema y a su vez haya elegido a otro usuario con el que comunicarse.

4.1.6 Requisitos de Interfaz de Usuario:

RIT-001: Interfaz Sencilla: La interfaz gráfica estará formada por un conjunto de ventanas, con el objetivo de que sea intuitiva y de fácil uso a los usuarios.

RIT-002: Mensajes Alerta: Los mensajes de alerta que el sistema emita corresponderán a la información de activación, o salida del sistema de videoconferencia. También se informará al usuario del inicio de una videoconferencia a través de la emisión de un sonido, tipo telefonía normal.

4.2 ESQUEMA DE DESARROLLO DE MÓDULO:

Una vez identificados los requisitos del sistema a desarrollar y en relación con el mayor cumplimiento posible de los mismos se propone el siguiente esquema de actividades y la secuenciación a nivel general del estado de presencia en el entorno de videoconferencia.

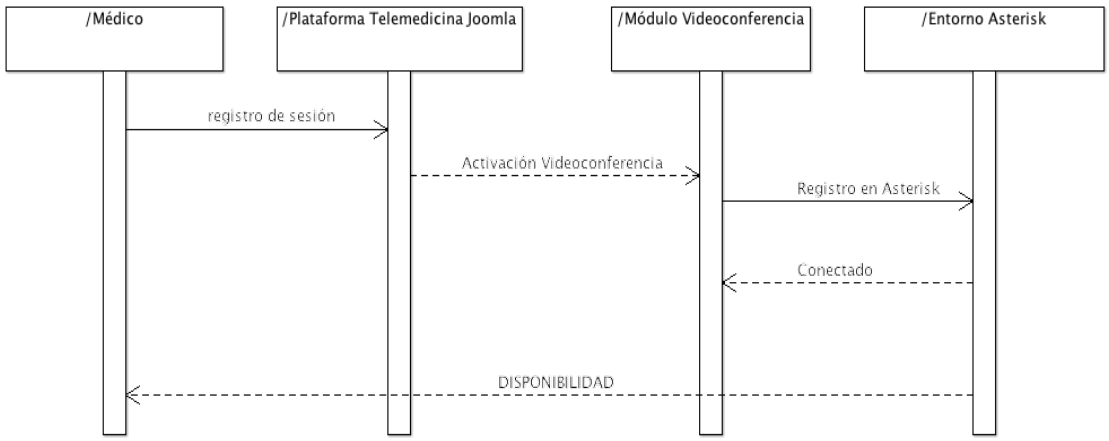


Fig. 3 Diagrama General de secuenciación de ingreso, registro y activación de sesión del modulo de Videoconferencia dentro de la Plataforma de Telemedicina Rural.

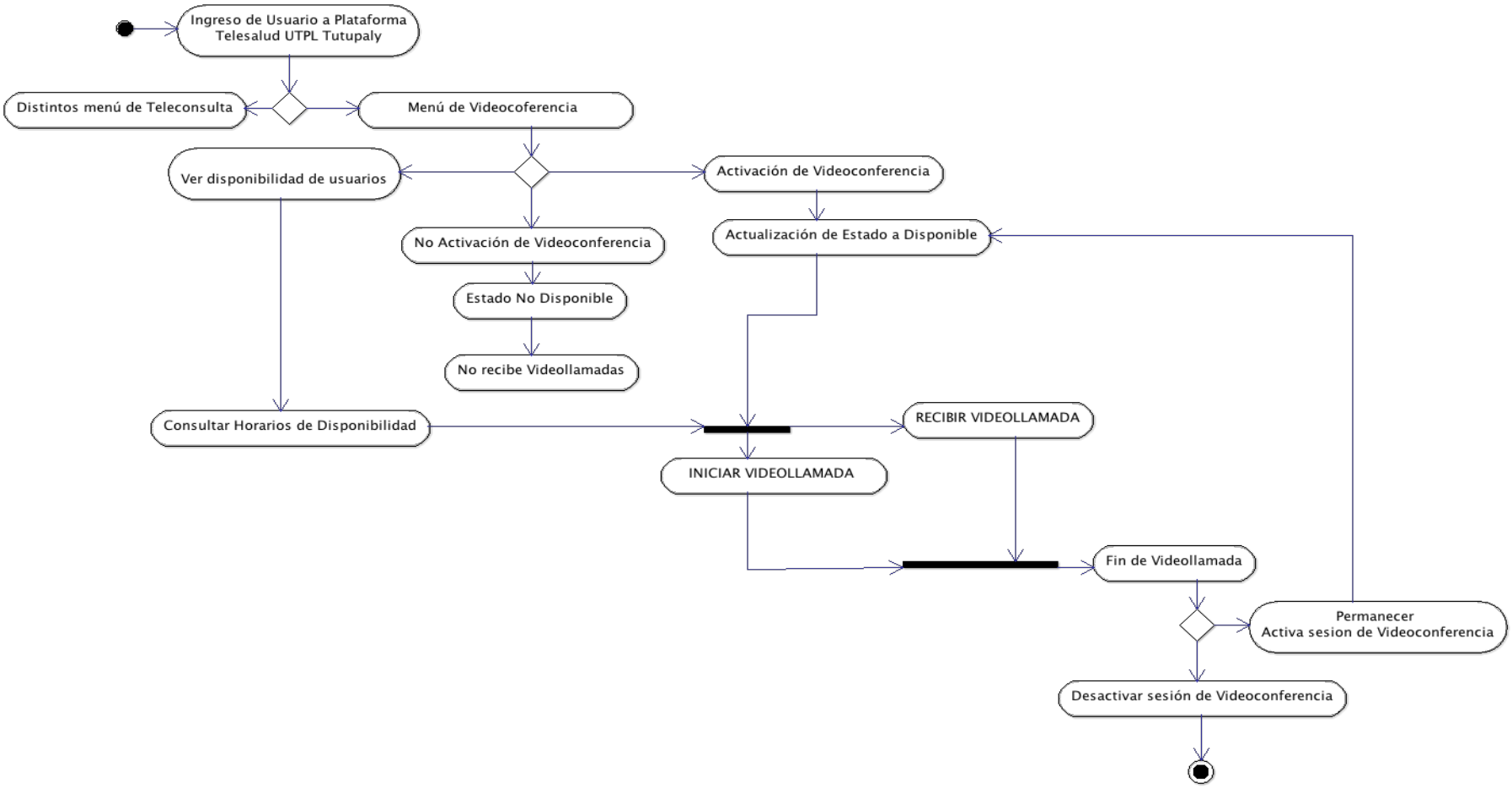


Fig. 4 Diagrama de Actividades Generales del módulo de Videoconferencia para Teleconsulta Directa

4.3 HERRAMIENTAS UTILIZADAS:

Como se había indicado anteriormente el presente módulo de videoconferencia será desarrollado e implementado de acuerdo con los requisitos identificados previamente y además siguiendo los lineamientos y modelo del sistema en el que se encuentra desarrollada la interfaz web de la Plataforma de Telemedicina del proyecto Tutupaly; esto es:

- Utilización del gestor de contenidos web Joomla 2.5.
- Gestor de base de datos MySQL.
- Lenguaje de programación y web PHP, HTML, JavaScript, JQuery.
- Servidor Apache instalado en Ubuntu 12.04 -Linux.
- Asterisk
- Sipml5
- Webrtc2sip

Debido a que el desarrollo e implementación del módulo de videoconferencia se realizará utilizando tecnología WebRTC, para tal efecto se hace uso del software libre *webrtc2sip* que actuará como pasarela para permitir el establecimiento de comunicación entre *Asterisk 11.9.0* y el entorno *sipml5*.

En las siguientes figuras se indican la arquitectura general del sistema de videoconferencia y su correspondiente diagrama de secuenciación de la señalización y transporte utilizados para la comunicación.

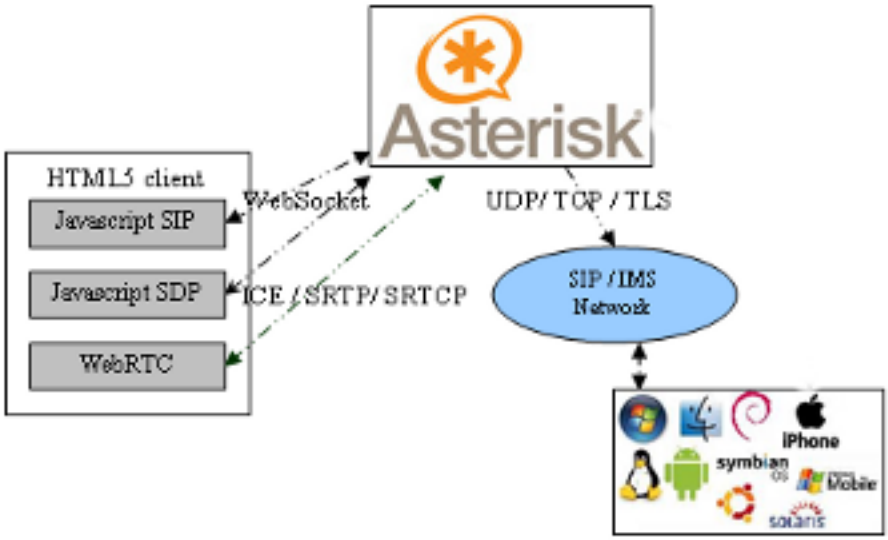


Fig. 5 Arquitecturas de Webrtc2sip utilizada para el diseño del módulo de videoconferencia [24]

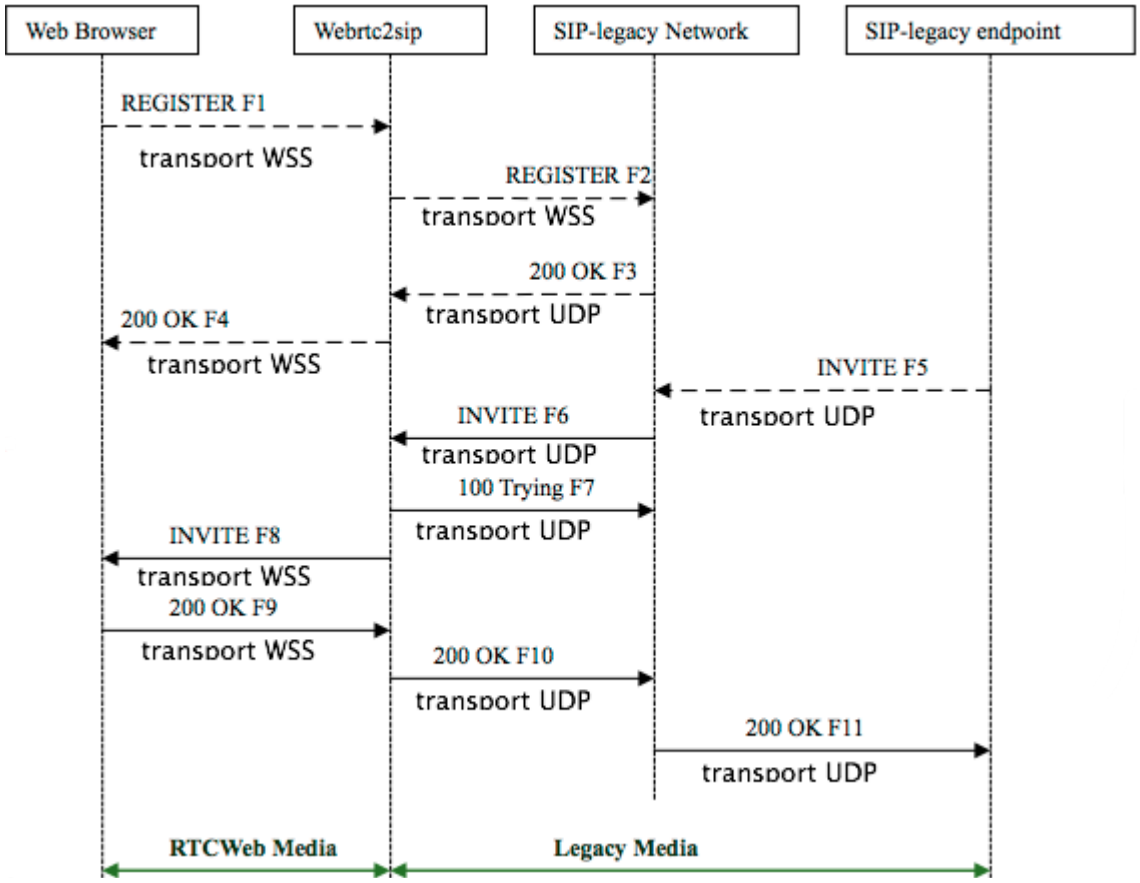


Fig. 6 Diagrama de secuenciación y señalización de protocolos de transporte a utilizar en el módulo de videoconferencia [24].

Adicional a lo anterior es menester resaltar que la seguridad de los datos contenidos en el módulo de videoconferencia, seguirán la misma línea de seguridad planteada por la plataforma general.

5. RESULTADOS

5.1 MODELADO:

Antes del desarrollo del módulo de videoconferencia se realizará el modelado de los principales casos de uso de acuerdo con los requisitos identificados en la metodología. A continuación en la siguiente tabla se presentan los principales actores del sistema a desarrollar en conjunto con sus actividades principales de caso de uso.

Tabla II. Actores del sistema y sus correspondientes casos de uso

Actor	Descripción	Caso de Uso
Administrador	Se encarga de las operaciones generales y específicas del módulo de videoconferencia, a las que ningún otro usuario tiene acceso.	➢ Crear usuarios ➢ Dar de baja usuarios ➢ Resolver problemas al establecer comunicación.
Médico Rural	Es la persona que se encuentra en un entorno rural o alguien que requiere de la consulta en directo de un médico especialista.	• Ingreso a la Plataforma • Ingreso al módulo de Videoconferencia • Activación de Videoconferencia • Selección de médicos especialistas. • Identificación de personas disponibles para videoconsulta. • Consultar calendario y horario de disponibilidad de especialistas. • Realizar videollamadas • Recibir videollamadas. • Desactivar videoconferencia
Médico Especialista	Es la persona que se encuentra por lo general en un centro de especialidades y atiende videoconsultas de médicos rurales u otros.	• Ingreso a la Plataforma • Ingreso al módulo de Videoconferencia • Activación de Videoconferencia • Selección de médicos rurales. • Identificación de personas disponibles para videoconsulta.

		• Ingreso de Horario de disponibilidad. • Consultar calendario y horario de disponibilidad. • Realizar videollamadas. • Recibir videollamadas. • Desactivar videoconferencia
Directivo/administrativo	Es la persona que gestiona el personal médico rural y especialista.	• Ingreso a la Plataforma • Ingreso al módulo de Videoconferencia • Activación de Videoconferencia • Selección de médicos rurales. • Selección de médicos especialistas. • Identificación de personas disponibles para videoconsulta. • Consultar calendario y horario de disponibilidad. • Realizar videollamadas a Especialistas y Rurales. • Desactivar videoconferencia

5.1.1 Diagramas de casos de uso:

A continuación se presentan los diagramas de casos de uso que permitirían el modelado del sistema o módulo a desarrollar.

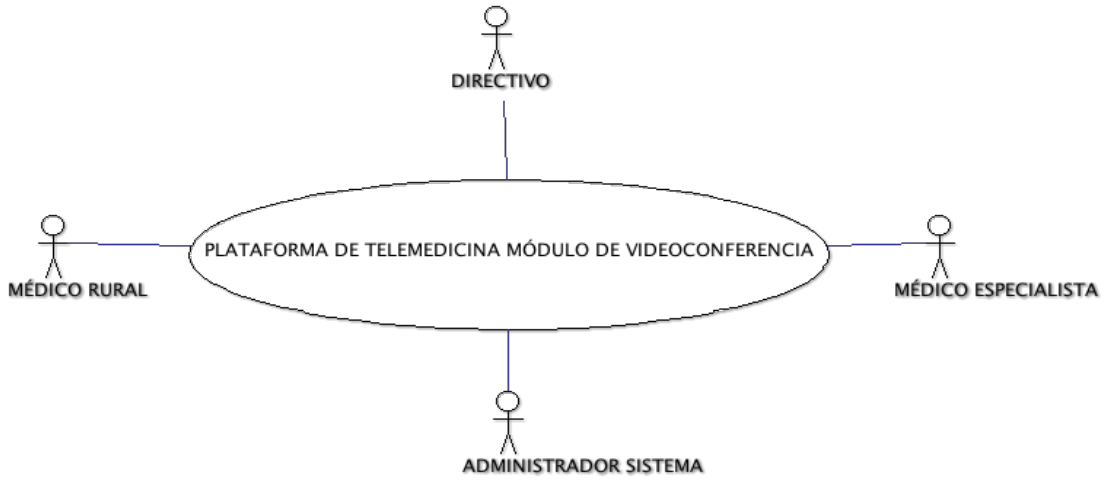


Fig. 7 Casos de Uso asociados al acceso del Módulo de Videoconferencia

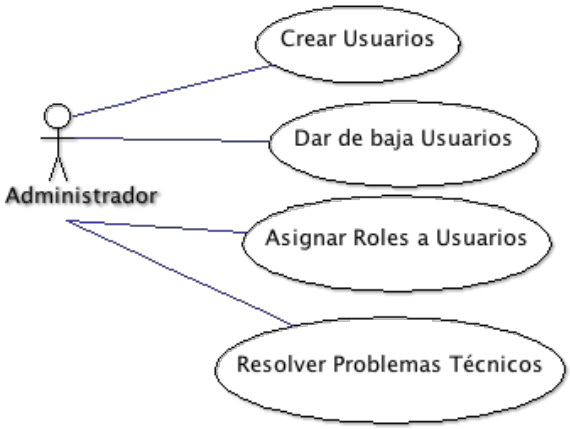


Fig. 8 Casos de Uso asociados a Gestionar Módulo

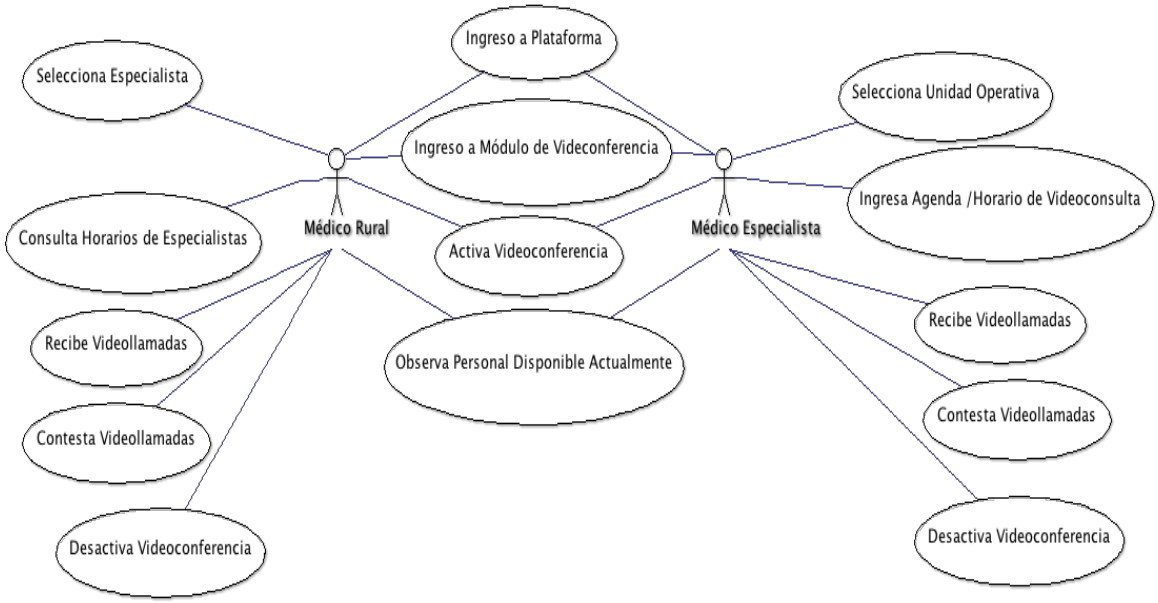


Fig. 9 Casos de Uso asociados a usuarios principales del módulo: Médico Rural y Especialista.

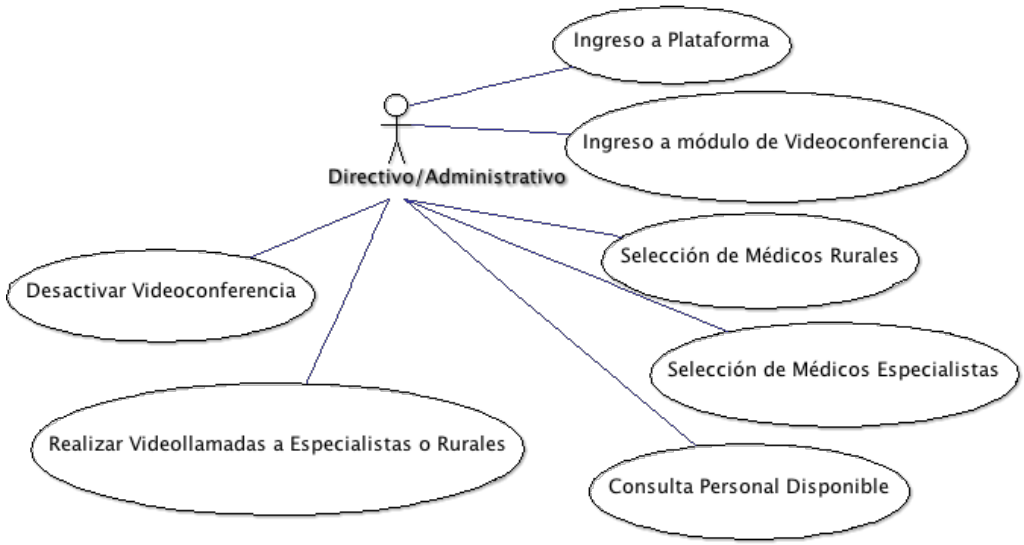


Fig. 10 Casos de Uso asociados a personal directivo/administrativo

5.2 DESARROLLO E IMPLEMENTACIÓN:

Para el desarrollo de la aplicación del módulo de videoconferencia se ha seguido un determinado proceso, que conlleva desde la identificación clara del modelado de clase y los casos de uso, hasta la instalación inicial de los sistemas operativos en los que se va a trabajar para finalmente diseñar el entorno o interfaz de interacción directa con el usuario.

A continuación se describe y detalla cada uno de los pasos a seguir en la implementación final del módulo.

5.2.1 Consideraciones y requerimientos pre-iniciales:

1. Se parte de la apreciación de que se tiene ya diseñada la plataforma de Telemedicina Rural [27], y por tanto; contiene información y características propias tanto de diseño como de servicios y prestaciones definidas; de ahí que el módulo a diseñar tendrá que adaptarse a esta plataforma ya indicada. En la figura siguiente se indica el diagrama esquemático presentado seguido para el diseño de la plataforma general.

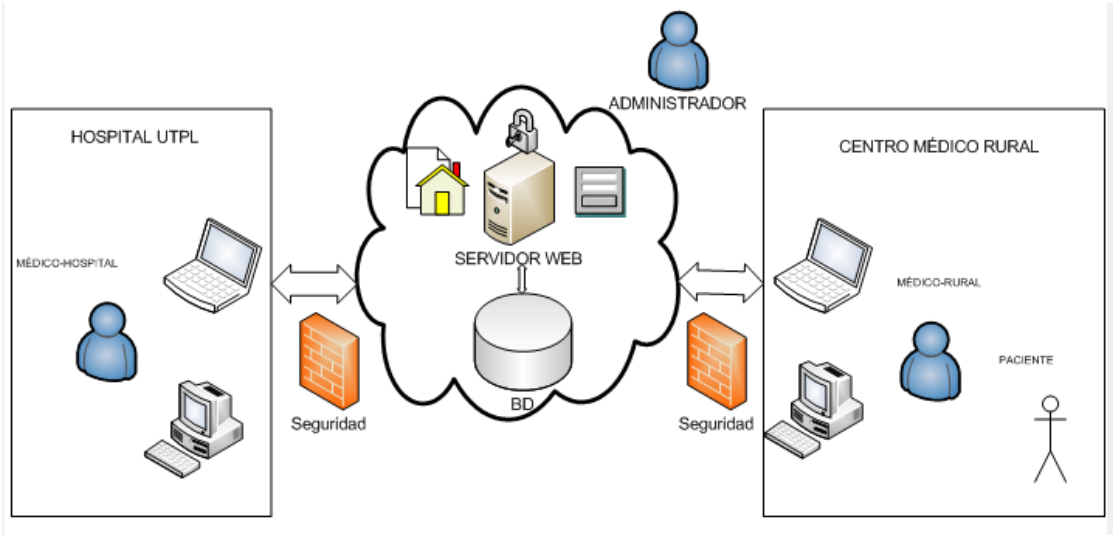


Fig. 11Diagrama Esquemático de la Plataforma General de Telemedicina Rural [27]

Telesalud UTPL
TUTUPALY
Pangui - Loja - Yacuambi
UNIVERSIDAD TÉCNICA PARTICULAR DE LOJA
ENTRAR
Usted está aquí: INICIO
! BIENVENIDO !
PLATAFORMA DE TELEMEDICINA DEL PROYECTO "TELESALUD UTPL TUTUPALY"
Si usted es parte del proyecto puede ingresar haciendo click en el botón: ENTRAR.
Si quiere saber más acerca del proyecto puede seguir el siguiente enlace: http://blogs.utpl.edu.ec/blogtutupaly/
Telesalud UTPL
TUTUPALY
Pangui - Loja - Yacuambi
UTPL Copyright © 2012. All Rights Reserved.
ING. LEONEL VASQUEZ DISEÑO & CONFIGURACIÓN

Fig. 12 Interfaz Principal de ingreso a la Plataforma General de Telemedicina Rural [27].

2. En la plataforma general se tiene diferentes grupos de usuarios, asignados a distintos roles distinguiéndoles entre si, de médicos especialistas, rurales y directivos. De esta forma se tiene un panel de menú distinto para cada usuario de acuerdo con su rol. En el caso del módulo de Videoconferencia o Teleconsulta Directa, estará disponible para todos los usuarios bajo un mismo nombre de acceso al menú.

5.2.2 Pasos iniciales e instalación de pasarela webrtc2sip:

Debido al requerimiento de comunicación multimedia, es necesario el tener un equipo con ciertas prestaciones mínimas de manera que al actuar como servidor y a su vez como lugar de ejecución de la pasarela de comunicación SIP no de lugar a problemas de congelamiento o lentitud en su tiempo de procesamiento. Esto es de suma importancia por cuanto se realizó pruebas en un equipo con poca cantidad de memoria RAM y esto dio lugar a inconvenientes relacionados principalmente con el audio y video.

El equipo que se ha utilizado para este proyecto contiene las siguientes características:

- Sistema Operativo: Ubuntu 12.04 LTS
- Memoria Ram: 4GB
- Tipo de SO: 32 bits.
- Procesador: Intel Core i5-2320 CPU @ 3.00GHzx4
- Disco: 200,1 GB

Definidos las prestaciones del equipo a trabajar se procede a instalar los componentes necesarios para el establecimientos de la pasarela webrtc2sip. Estos componentes son:

- ✓ Instalación servidor web Apache 2.0
- ✓ Instalación software Asterisk 11.9.0
- ✓ Instalación software webrtc2sip a través de la guía técnica proporcionada por Doubango Telecom [24-Anexo 2] .
- ✓ Adicional a lo anterior también se instala los gestores de base de datos de MySQL y paquetes de librerías php.

5.2.3 Configuración de pasarela webrtc2sip:

Una vez instalados estos componentes y realizadas las configuraciones necesarias de acuerdo con las indicaciones técnicas y tutoriales, se procede a la configuración de *sipml5*, esto también se lo realiza en la medida de que si surgieren problemas, se puede ayudar de preguntas y respuestas a foros web propios de cada componente.

En la siguiente figura se indica la configuración respectiva y definitiva referente a *sipml5* en cuanto se refiere a la designación y dirección de los puertos a utilizar en la pasarela de webrtc2sip.

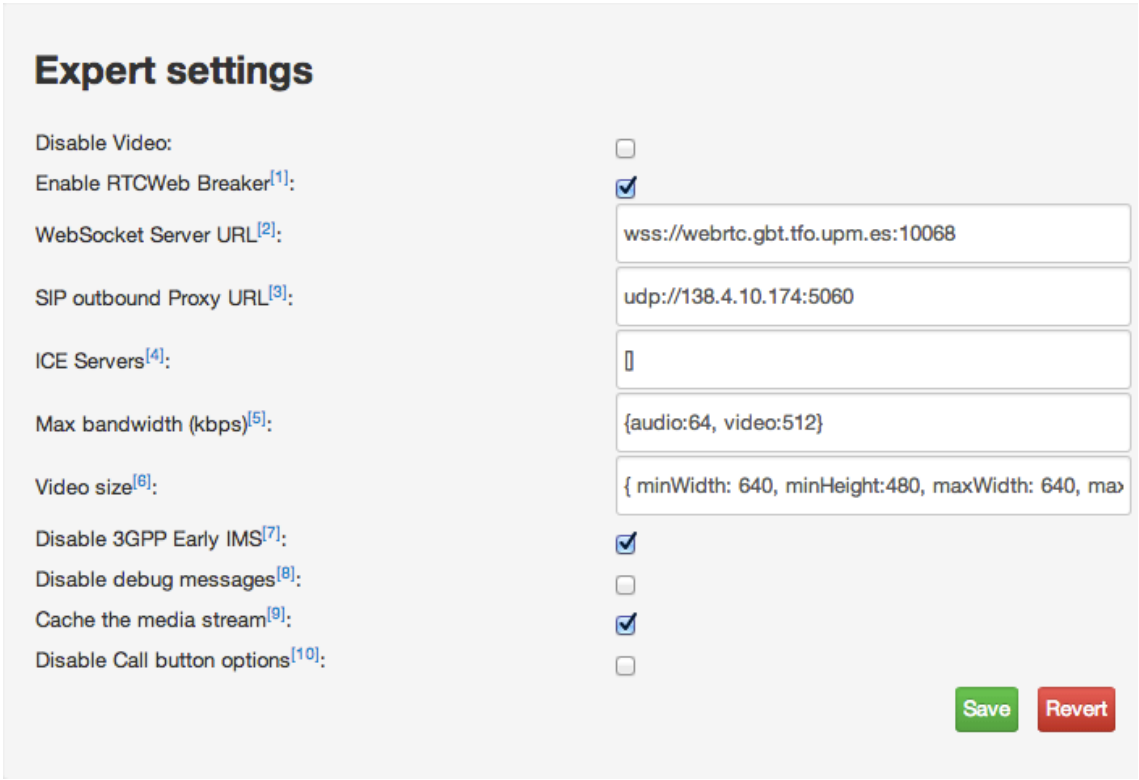
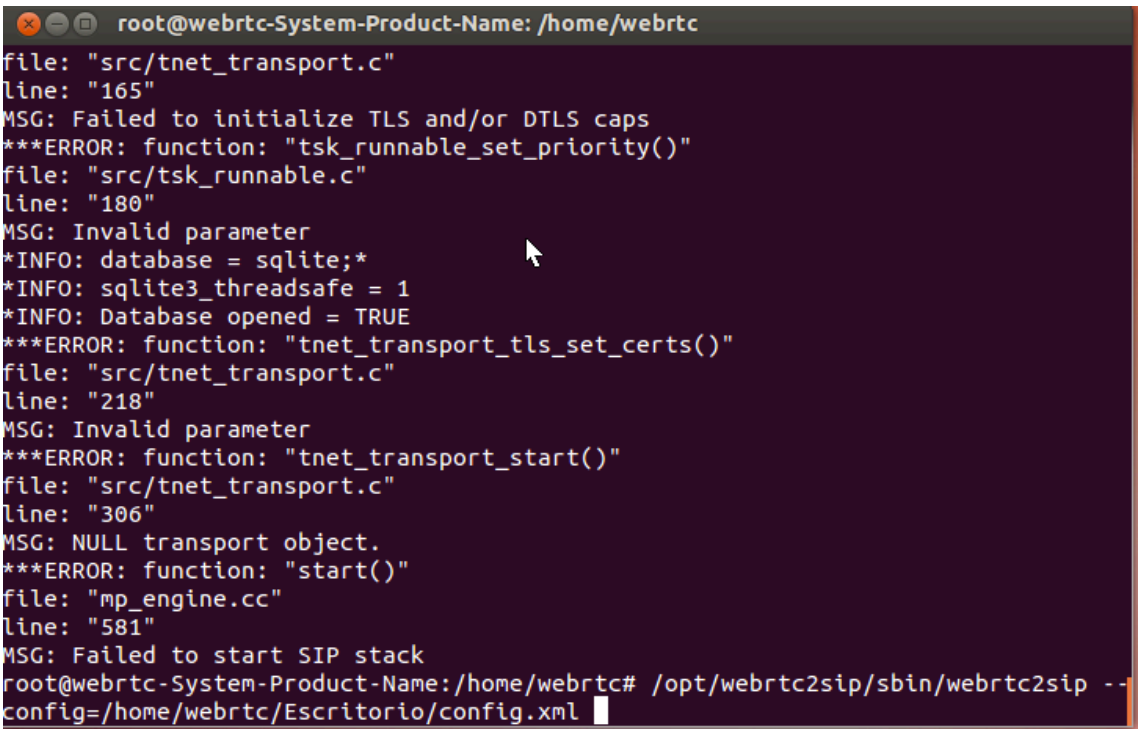


Fig. 13 Configuración de puertos a ser utilizados por la pasarela webrtc2sip.



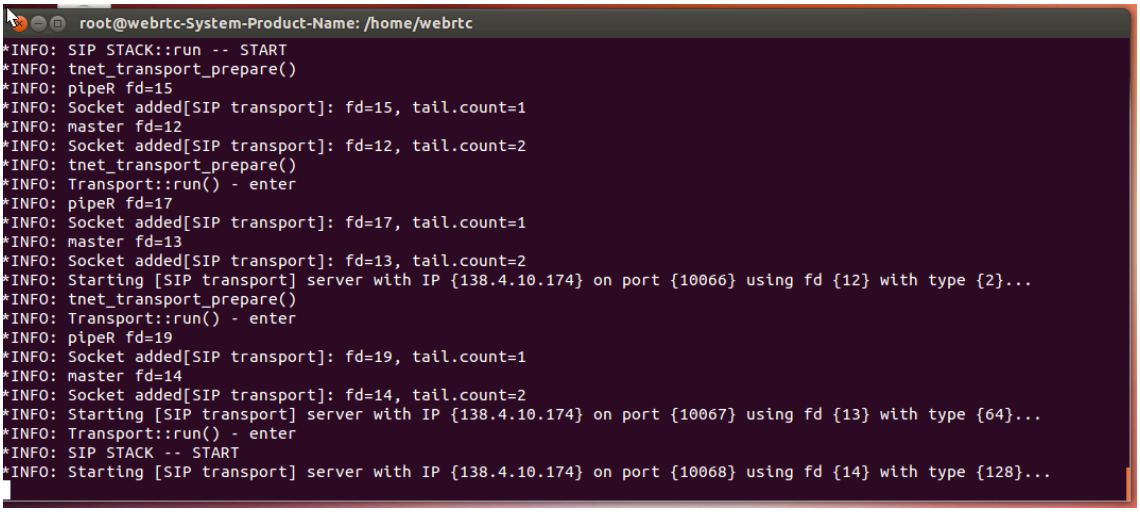

root@webrtc-System-Product-Name: /home/webrtc
*INFO: SIP STACK::run -- START
*INFO: tnet_transport_prepare()
*INFO: pipeR fd=15
*INFO: Socket added[SIP transport]: fd=15, tail.count=1
*INFO: master fd=12
*INFO: Socket added[SIP transport]: fd=12, tail.count=2
*INFO: tnet_transport_prepare()
*INFO: Transport::run() - enter
*INFO: pipeR fd=17
*INFO: Socket added[SIP transport]: fd=17, tail.count=1
*INFO: master fd=13
*INFO: Socket added[SIP transport]: fd=13, tail.count=2
*INFO: Starting [SIP transport] server with IP {138.4.10.174} on port {10066} using fd {12} with type {2}...
*INFO: tnet_transport_prepare()
*INFO: Transport::run() - enter
*INFO: pipeR fd=19
*INFO: Socket added[SIP transport]: fd=19, tail.count=1
*INFO: master fd=14
*INFO: Socket added[SIP transport]: fd=14, tail.count=2
*INFO: Starting [SIP transport] server with IP {138.4.10.174} on port {10067} using fd {13} with type {64}...
*INFO: Transport::run() - enter
*INFO: SIP STACK -- START
*INFO: Starting [SIP transport] server with IP {138.4.10.174} on port {10068} using fd {14} with type {128}...

Fig. 14 Indicación de inicio de pasarela webrtc2sip y correcto funcionamiento de la misma.

5.2.4 Ensayos y pruebas de funcionamiento de pasarela:

Concluidas las configuraciones respectivas, se procede a la realización de pruebas de ensayo. Para ello se utilizará dos clientes de prueba correspondientes a 101 y 104, cliente que inicia la videoconferencia y cliente que la acepta y recibe; respectivamente.

Seguidamente se indican algunas imágenes de las pruebas del funcionamiento de la pasarela y a su vez el establecimiento de la comunicación multimedia a través de WebRTC.

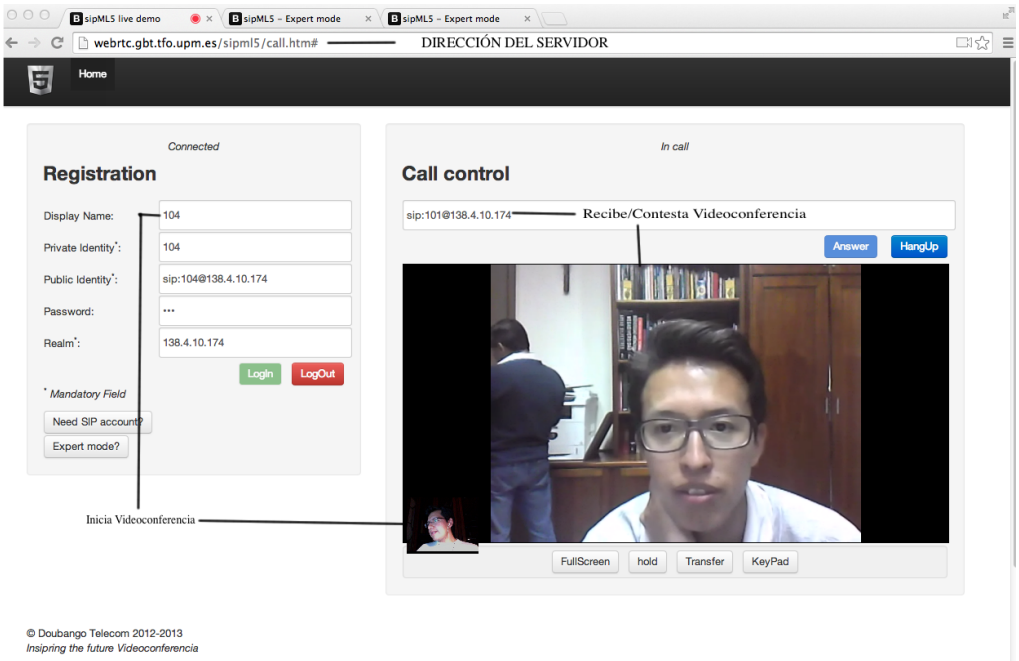


Fig. 15 Pruebas de ensayo de establecimiento de conexión de videoconferencia a través de WebRTC

5.3 DESARROLLO DE INTERFAZ DE MÓDULO DE VIDEOCONFERENCIA E INTEGRACIÓN CON *WEBRTC2SIP:*

A la par que se ha ido trabajando en el desarrollo de la pasarela de comunicación SIP. Se realiza el diseño del módulo de videoconferencia dentro del entorno de la Plataforma General de Teleconsultas, tomando en cuenta las consideraciones pre-iniciales manifestadas en la sección correspondiente; esto es especialmente el diseño en el software *Joomla 2.5*.

Debido a que el módulo de videoconferencia será una parte de la Plataforma General de Telemedicina, se ha establecido el incrementar una sección en los menús principales de los diferentes servicios ofrecidos a los distintos usuarios.

Dicha sección se denominará *"Videoconferencia"* y estará en la parte superior de la sección de menú principal. La siguiente figura indica lo señalado.

Telesalud UTPL
TUTUPALY
Panguí - Loja - Yacuambi
SALIR
Error conectando a la base de datos.
HOLA: Irene Carrillo
PRESENTACION TELECONSULTAS RECIBIDAS GENERAR RESPUESTA ENVIO DE ERRORES TELECONSULTA DIRECTA VIDEOCONFERENCIA
Usted está aquí: INICIO
Sección de Ménu de Módulo de Videoconsulta
! BIENVENIDO !
PLATAFORMA DE TELEMEDICINA DEL PROYECTO "TELESALUD UTPL TUTUPALY"
Si usted es parte del proyecto puede ingresar haciendo click en el botón: ENTRAR.
Si quiere saber más acerca del proyecto puede seguir el siguiente enlace: http://blogs.utpl.edu.ec/blogtutupaly/
Adjuntos:
Instrucciones de Acceso Especialistas [Instrucciones de Acceso Especialistas] 2481 Kb
UTPL Copyright © 2012. All Rights Reserved.

Fig. 16 Pestaña del módulo de Videoconsulta en la plataforma Telesalud UTPL Tutupaly

Una vez realizada la elección del menú de Videoconferencia destinada para el módulo, se procede de acuerdo con los requerimientos identificados al diseño de la interfaz de dicho módulo.

Se ha considerado el diseño de una interfaz en apariencia similar para todos los usuarios, no obstante, difieren entre si en funcionalidad en relación con el rol de usuario o grupo al que pertenezca el médico que ha ingresado al sistema. Así se tiene:

- ***Interfaz de médico rural:*** la cual permitirá al usuario activar la videoconferencia o su estado, consultar los usuarios disponibles de manera directa y si requiere como es el caso de estos de una sección personalizada de búsqueda de los especialistas disponibles y su correspondiente horario.

- ***Interfaz de médico especialista:*** permite activar la videoconferencia o estado, consultar los usuarios disponibles de manera directa y si búsqueda personalizada de los médicos rurales disponibles.
- ***Interfaz de directivo:*** es una combinación de los dos anteriores, permitiendo la consulta personalizada de especialistas y médicos rurales disponibles , al igual que la correspondiente presentación de usuarios disponibles en el instante.

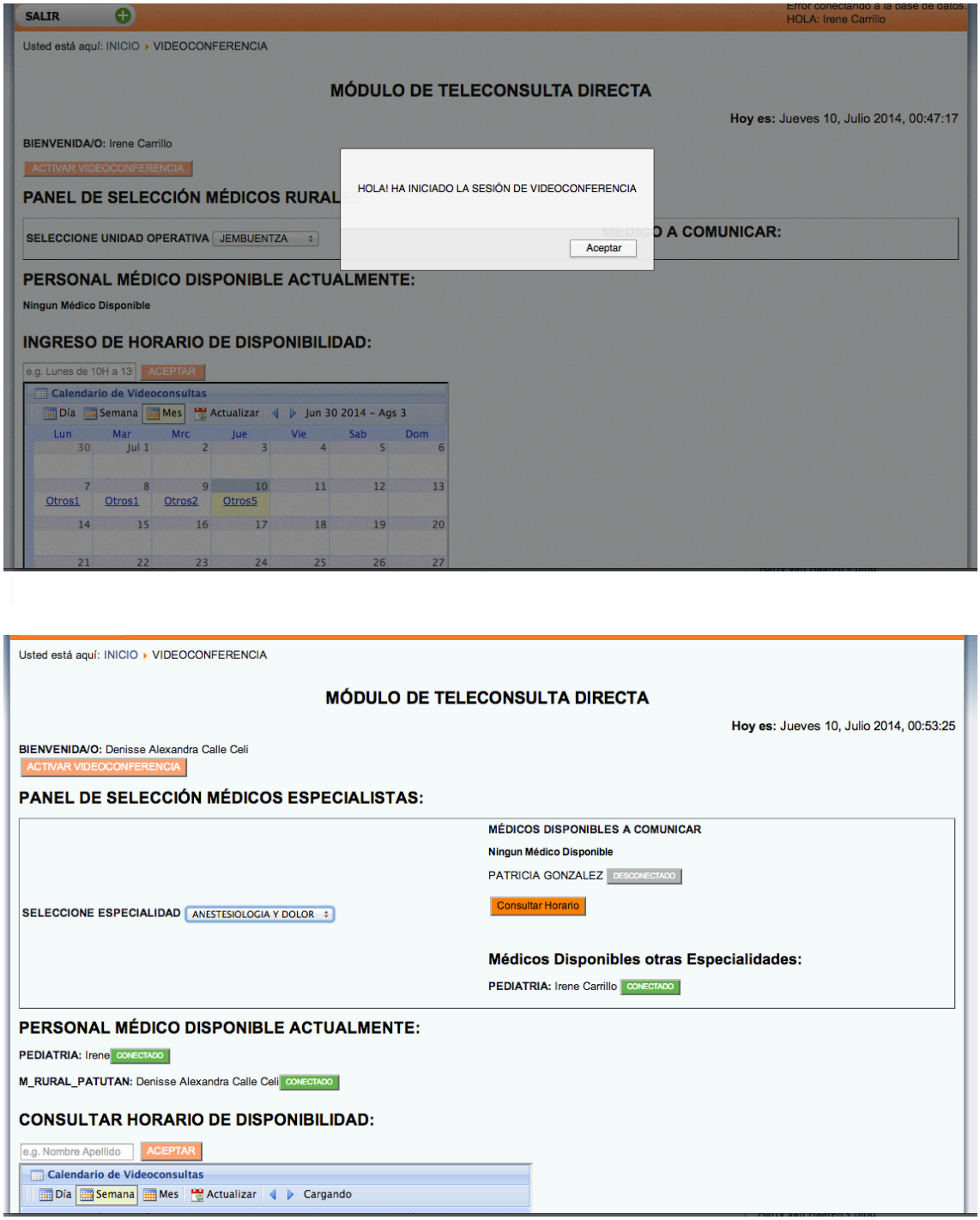

SALIR
HOLA: Irene Carrillo
Usted está aquí: INICIO ▸ VIDEOCONFERENCIA
MÓDULO DE TELECONSULTA DIRECTA
Hoy es: Jueves 10, Julio 2014, 00:47:17
BIENVENIDA/O: Irene Carrillo
ACTIVAR VIDEOCONFERENCIA
HOLA! HA INICIADO LA SESIÓN DE VIDEOCONFERENCIA
Aceptar
PANEL DE SELECCIÓN MÉDICOS RURAL
SELECCIONE UNIDAD OPERATIVA JEMBUENTZA
A COMUNICAR:
PERSONAL MÉDICO DISPONIBLE ACTUALMENTE:
Ningun Médico Disponible
INGRESO DE HORARIO DE DISPONIBILIDAD:
e.g. Lunes de 10H a 13
ACEPTAR
Calendario de Videoconsultas
Día Semana Mes Actualizar Jun 30 2014 – Ags 3
Lun Mar Mrc Jue Vie Sab Dom
Otros1 Otros1 Otros2 Otros5
Usted está aquí: INICIO ▸ VIDEOCONFERENCIA
MÓDULO DE TELECONSULTA DIRECTA
Hoy es: Jueves 10, Julio 2014, 00:53:25
BIENVENIDA/O: Denisse Alexandra Calle Celi
ACTIVAR VIDEOCONFERENCIA
PANEL DE SELECCIÓN MÉDICOS ESPECIALISTAS:
MÉDICOS DISPONIBLES A COMUNICAR
Ningun Médico Disponible
PATRICIA GONZALEZ DESCONECTADO
SELECCIONE ESPECIALIDAD ANESTESIOLOGIA Y DOLOR
Consultar Horario
Médicos Disponibles otras Especialidades:
PEDIATRIA: Irene Carrillo CONECTADO
PERSONAL MÉDICO DISPONIBLE ACTUALMENTE:
PEDIATRIA: Irene CONECTADO
M_RURAL_PATUTAN: Denisse Alexandra Calle Celi CONECTADO
CONSULTAR HORARIO DE DISPONIBILIDAD:
e.g. Nombre Apellido
ACEPTAR
Calendario de Videoconsultas
Día Semana Mes Actualizar Cargando

Fig. 17 Interfaz del Módulo de Videoconferencia para médicos especialistas y médicos rurales, respectivamente.

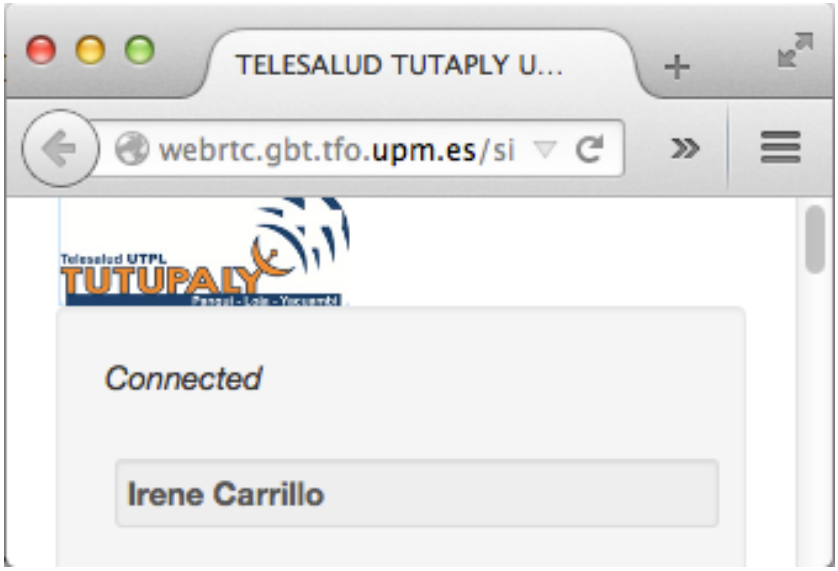
Para desactivar el módulo de videoconferencia y desaparecer su presencia como "conectado" del módulo, simplemente se tiene que cerrar la ventana inicial de activación. Se presentará el siguiente mensaje.

Fig. 18 Mensaje informativo al desactivar el módulo de Videoconferencia

5.4 MÓDULO DE VIDEOCONFERENCIA EN FUNCIONAMIENTO

Realizados los diseños respectivos de las interfaces de los distintos usuarios de acuerdo con los roles, el siguiente paso es integrar estos con la funcionalidad de WebRTC para de esta manera poder establecer la comunicación de videoconferencia.

Esto se lo hace a través de la activación del servicio de establecimiento de comunicación, a través del botón de *"Activar Videoconferencia"* indicado en la interfaz principal del usuario, de esta manera permitirá a su vez indicar la disponibilidad y presencia del mismo para que pueda recibir o realizar videoconsultas. En la siguiente figuras se indica la ventana de confirmación de que se ha conectado correctamente el usuario/médico.



En el instante que un médico decide realizar una videoconsulta, la página del navegador le *solicitará el permiso para poder acceder a las funcionalidades de la cámara y micrófono de su computador*; lo cual es algo normal y de seguridad en la utilización de tecnología WebRTC, como es en el presente caso. Esta petición también le sucederá a la persona con la que se quiera comunicar. La notificación o aviso para aceptar una videoconferencia es a través de un sonido.

En las siguientes figuras se describe un ejemplo, para probar la funcionalidad del módulo de videoconsulta. Se sigue el siguiente proceso:

El usuario identificado como médico rural (Denisse Calle) intenta establecer una videoconsulta con la especialista en Pediatría (Irene Carrillo), quienes a su vez ya han activado su cuenta de videoconferencia; por tanto, luego de que la médico rural seleccione a la especialista para la comunicación, se apreciará las solicitudes del acceso a la cámara y micrófono a través del navegador web. Concedido el permiso a la petición anterior, se presentaran las opciones de responder o rechazar la video llamada, a la par que se está produciendo o emitiendo un sonido de aviso, cual si fuese un teléfono convencional, todo esto para el caso de la médico especialista. El tiempo en que declina una videollamada, si no recibiere contestación es de 18 segundos. El siguiente esquema y figuras indican a nivel de detalle lo descrito.

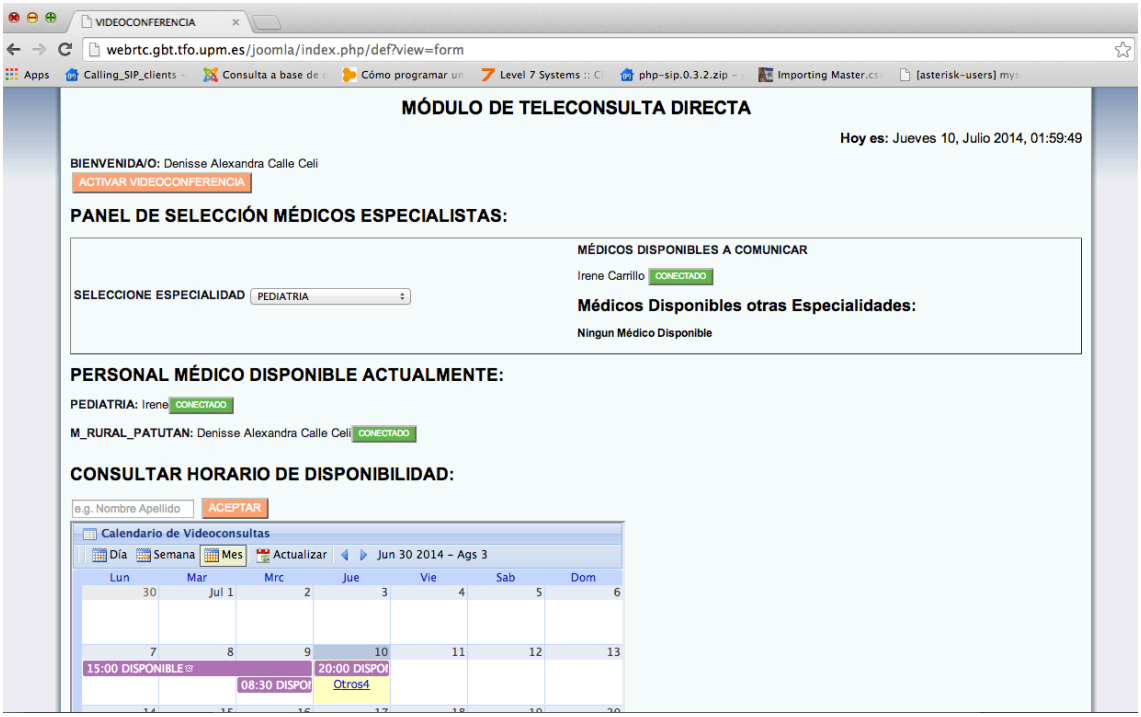


Fig. 19 Selección de médico especialista a comunicar en la interfaz de médico rural

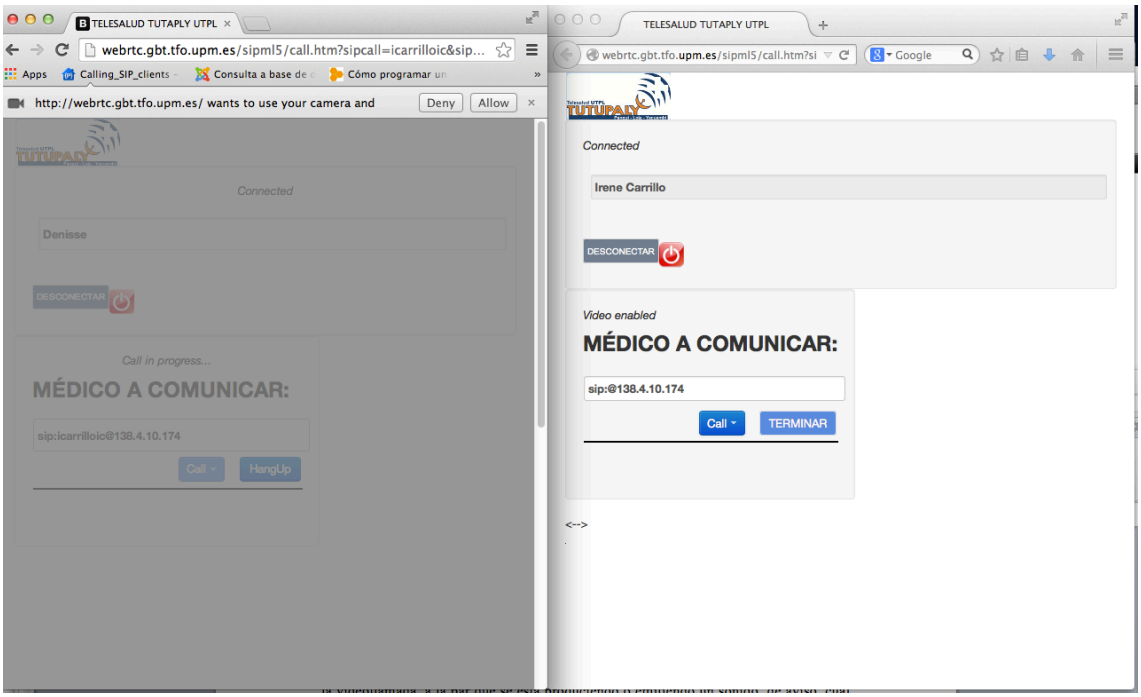


Fig. 20 Petición de acceso a cámara y micrófono de parte de quien realiza la video llamada

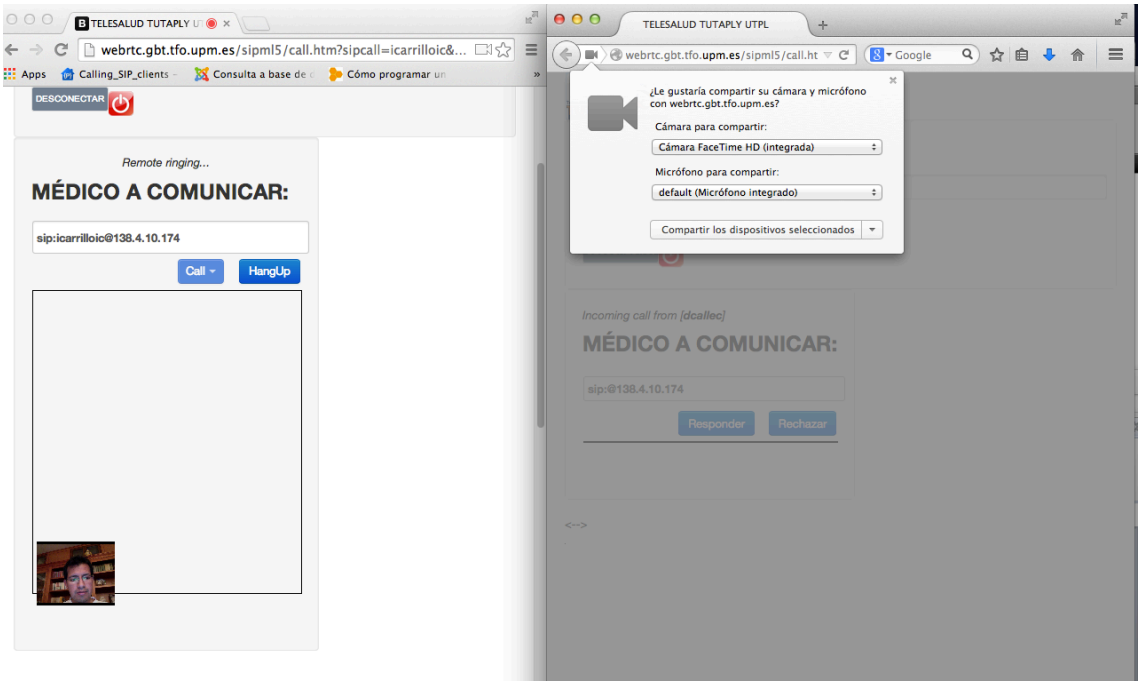


Fig. 21 Petición de acceso a cámara y micrófono a quien se realiza la video llamada y además informe o aviso de la llamada entrante a través de sonidos.

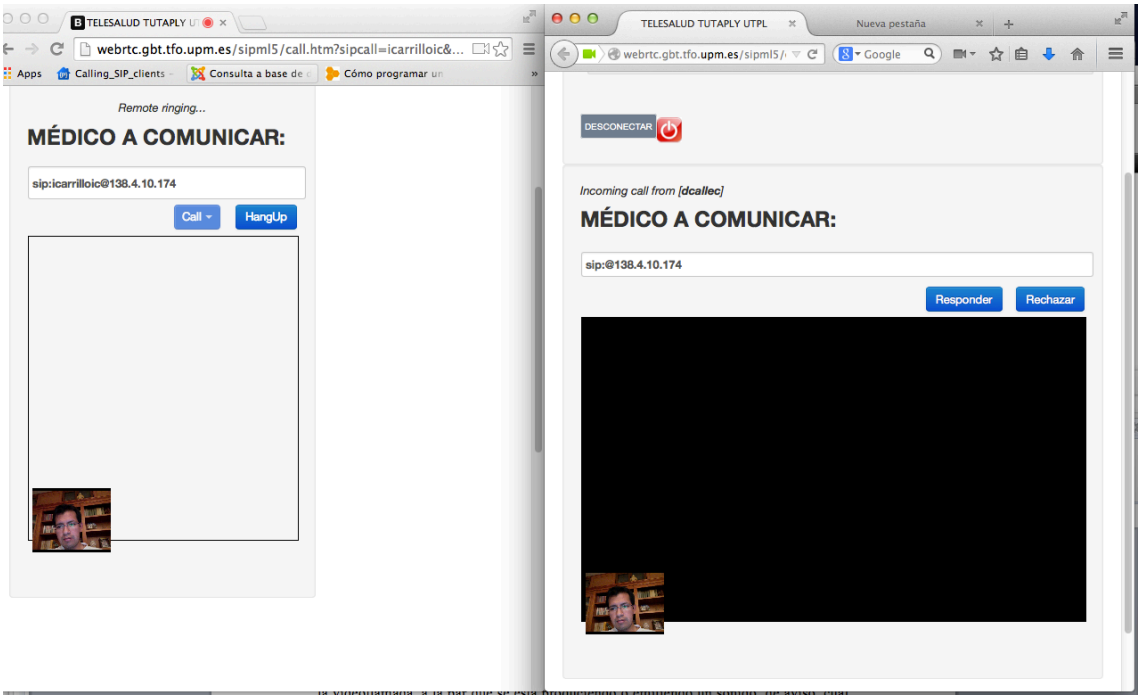


Fig. 22 Dados los permisos de acceso a cámara y micrófono, se da la posibilidad de responder o rechazar la videollamada.

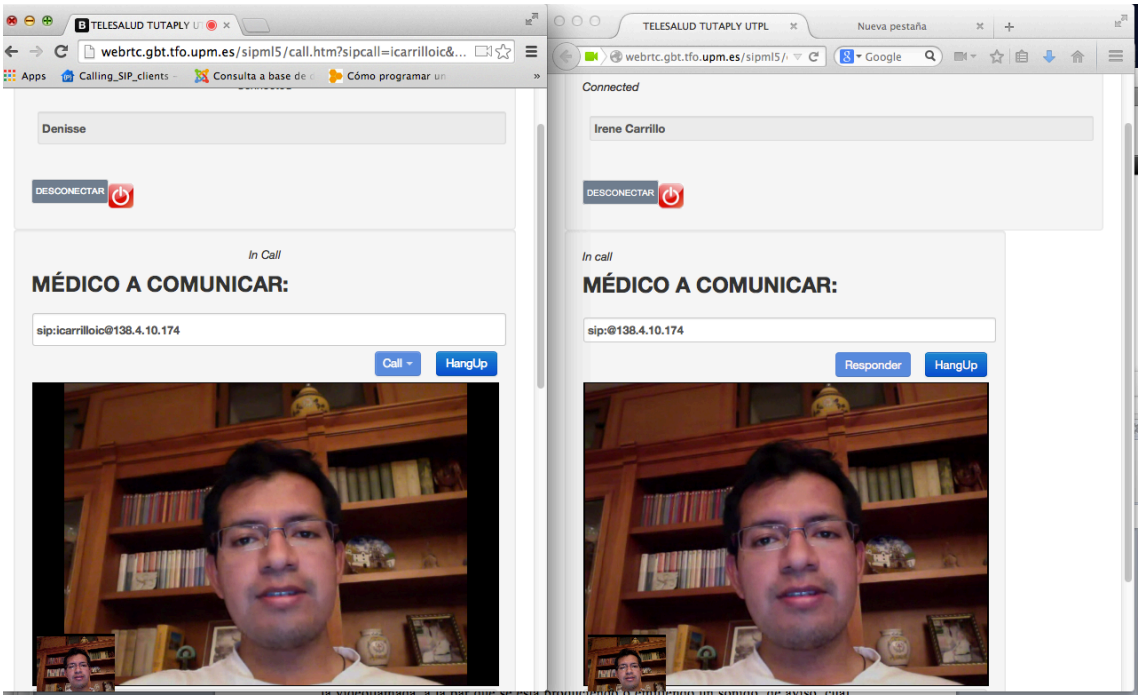


Fig. 23 Establecimiento de Videoconferencia entre usuario identificado como medico rural y especialista

5.4.1 Calendario de Disponibilidad de Médicos Especialistas

Adicional a la videoconferencia desarrollada en la interfaz del usuario se le da la posibilidad al médico Especialista de crear y poner en un calendario su horario de disponibilidad de acuerdo con los días y horas en los que crea conveniente. Este calendario es de consulta también para los médicos rurales, en caso de que quisieren ver el horario de un especialista en específico. En las siguientes figuras se indican las interfaces correspondientes para médicos especialistas y médicos rurales.

Fig. 24 Aspecto de Calendario de Médico Especialista.

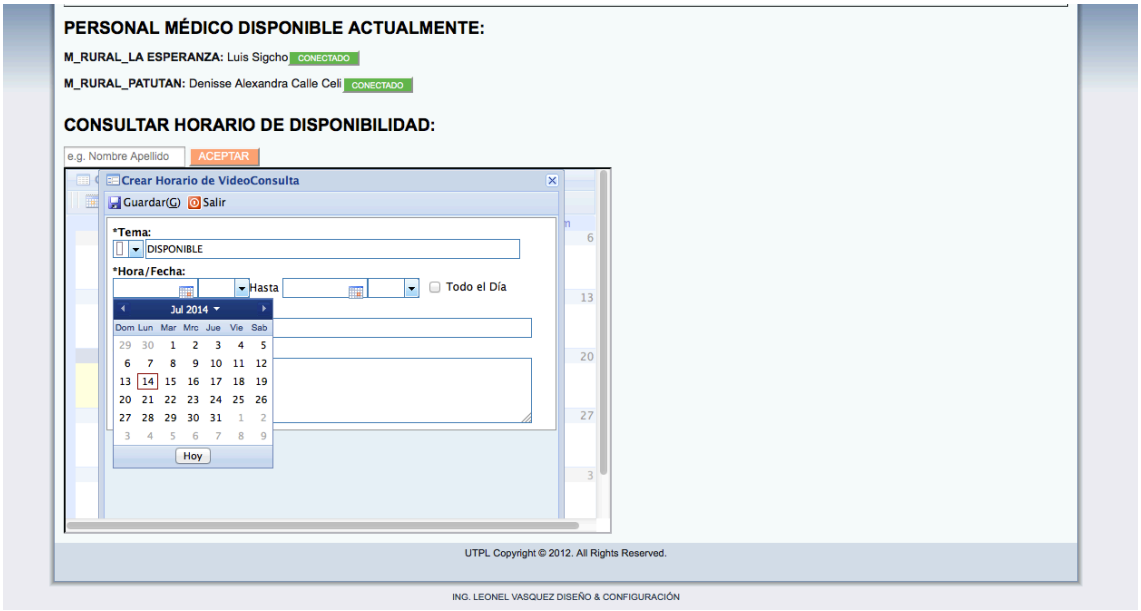


Fig. 25 Ingreso de Horario de disponibilidad de Calendario de Médico Especialista.

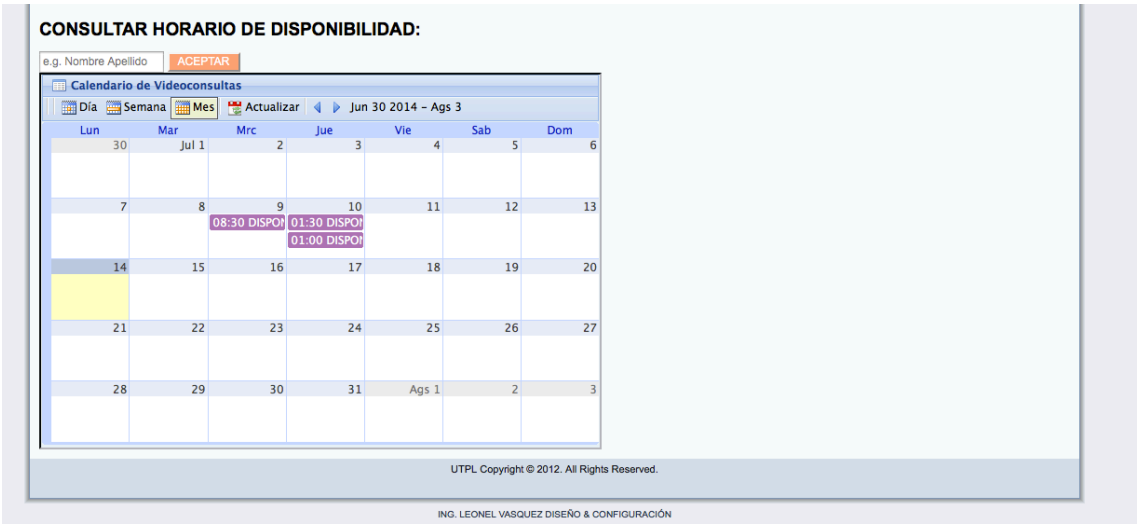


Fig. 26 Aspecto de Calendario de Médico Rural para consulta de horario de médicos Especialistas.

6. CONCLUSIONES

En el presente trabajo se ha propuesto el desarrollo e implementación de un módulo de videoconferencia para la plataforma de Telemedicina Rural del proyecto Telesalud UTPL Tutupaly de la Universidad Técnica Particular de Loja en colaboración con el Grupo de Bioingeniería y Telemedicina de la Universidad Politécnica de Madrid, por lo que al finalizarlo se concluye que:

- El módulo de teleconsulta desarrollado permitirá una mejora respecto a obtener una mayor ventaja en cuanto a la utilización de la plataforma rural ya diseñada; puesto que se incrementará la modalidad de teleconsulta directa y síncrona.

- El módulo propuesto al ser desarrollado a través de WebRTC, se convierte en una aplicación *open source*, lo cual es muy beneficioso para ser aplicado en todos los lugares, ya que al utilizar únicamente los navegadores web, no constituye una aplicación de instalación, sino únicamente requiere de la conexión a la web.

- La creación del módulo de videoconferencia, permitirá dar un plus a las distintas modalidades de teleconsulta y a su vez tener una mayor implicación en cuanto se refiere a la atención de los pacientes y a la formación de los médicos rurales.

- Debido a que la tecnología WebRTC utilizada para el desarrollo del presente módulo, aún es reciente y estar en continua investigación y mejoramiento, tiene la limitación de garantizar su funcionamiento al 100%, únicamente con la utilización de los navegadores Mozilla Firefox y Google Chrome.

- El módulo de videoconferencia desarrollado, dado sus características de funcionalidad es factible de aplicarlo también en otros ámbitos distintos de la telemedicina, como es el caso para la formación o educación a distancia.

7. TRABAJOS FUTUROS

Con el objetivo de establecer mejoras adicionales al presente módulo de videoconferencia, se propone los siguientes trabajos futuros:

- ✓ Evaluación directa del módulo propuesto por parte de los usuarios de la plataforma de Telemedicina UTPL Tutupaly; con el fin de obtener realimentación referente a los aspectos de funcionalidad, complejidad en la utilización, efectividad, y usabilidad del sistema.

- ✓ Creación de una sección de informes y estadísticas detallados del servicio de videoconferencia, esto es correspondiente únicamente a los médicos directivos/administrativos.

- ✓ El establecimiento de un método de mayor autonomía, en cuanto se refiere a la gestión de usuarios a través del entorno Asterisk, de tal forma que exista una mayor gestión y rapidez de acuerdo con el creciente número de usuarios.

8. BIBLIOGRAFÍA

[1] Kanthraj GR. Classification and design of teledermatology practices: What dermatoses? Which technology to apply?. JEADV 2009; 23:865-875.

[2] High, W. *et al*. Assessment of the accuracy of low-cost store and forward teledermatology consultation. J Am Acad Dermatol 2000; 42 (5):776-783.

[3] D'Angelo, M.; López Cotti, C; Casas, I. Teledermatología y Dermatología comunitaria: estrategias complementarias para mejorar la acecesibilidad y calidad de la atención dermatológica. Dermatol Rev Mex 2013; 57:446-453.

[4] Perednia DA, Allen A. Telemedicine technology and clinical applications. JAMA. 1995; 273:483–488.

[5] Casanova, J; Buti, M; *et al*. Teledermatology. Med Cutan Iber Lat Am 2005; 33(2)53-64.

[6] Trondsen et al. VIDEOCARE: Decentralised psychiatric emergency care through videoconferencing. BMC Health Services Research 2012, 12:470-74.

[7] Kanthraj GR. A longitudinal study of consistency in diagnostic accuracy of teledermatology tools. Indian J Dermatol Venereol Leprol 2013; 79:668-78.

[8]Eminović, N; de Keizer, NF; Bindels, PJ; Hasman, A. Maturity of teledermatology evaluation research: A systematic literature review. Br J Dermatol 2007; 156:412-9.

[9]Warshaw EM, Hillman YJ, Greer NL, Hagel EM, MacDonald R, Rutks IR, et al. Teledermatology for diagnosis and management of skin conditions: A systematic review. J Am Acad Dermatol 2011; 64:759-72.

[10] Van der Heijden JP, Spuls PI, Voorbraak FP, de Keizer NF, Witkamp L, Bos JD. Tertiary teledermatology: A systematic review. Telemed J E Health 2010; 16:56-62.

[11] Smith, A.; White, M.; McBride, C. Multi-site videoconference tutorials for medical students in Australia. ANZ J Surg 2012; 82: 714–719.

[12] Hernando, M.E. Sistema de Teleasistencia en Centros Escolares. Grupo de Bioingeniería y Telemedicina. ETSIT. UPM 2001.

[13] Garshnek, V; Burkle, F. Applications of Telemedicine and Telecommunications to Disaster Medicine: Historical and Future Perspectives. J Am Med Inform Assoc 1999; 6: 26-37.

[14] Amante, H.; Pedreira, M.; Lung, C. Teledermatology: past, present and future. An. An. Bras. Dermatol. 2005; 80(5): 523-32.

[15] Viera, M.; Vivas, A.; Burdick, A. Teledermatología: Experiencia en Estados Unidos y Latinoamérica. Universidad de Miami, Escuela de Medicina Miller. [En Línea] Disponible en: antoniorondonlugo.com/blog/wp-content/.../122-Teledermatologia_.pdf

[16] De la Fuente, Justo. Médico Dermatólogo, Sacerdote. Entrevista sobre situación de Dermatología en Yaoundé_Camerún, diciembre 2013. (delafuentejusto@gmail.com)

[17] C.O. Bagayoko et al. Assessment of Internet-based tele-medicine in Africa (the RAFT project). Computerized Medical Imaging and Graphics 30 (2006) 407–416

[18] LeRouge, C.; Garfield, M.; Hevner, A. Quality Attributes in Telemedicine Video Conferencing. Proceedings of the 35th Annual Hawaii International Conference on System Sciences 2002 (HICSS-35-02).

[19] Smith, A.; Scuffham, P. Wootton, R. The Costs and potential savings of a novel telepediatric service in Queensland. BMC Health Services Research 2007, 7:35-42.

[20] N. Squibb. Video Transmission for Telemedicine. J Telemedicine and Telecare, 1999, p. 7

[21] Arcila, C.; Loaiza, M. Diseño de un enlace de telemedicine para el Hospital Universitario San Juan de Dios del Quindío. Universidad del Quindío. Armenia. 2010.

[22] Martinez Ruíz I. Contribuciones a modelos de tráfico y control de QoS en los nuevos servicios sanitarios basados en Telemedicina. Tesis doctoral, Departamento de Ingeniería Electrónica y Comunicaciones. Universidad de Zaragoza. Julio 2006

[23] WebRTC, [En Línea]. Disponible en: http://es.slideshare.net/Quobis/webinar-webrtc-y-html5-spanish-quobis.

[24] Doubango Telecom [En Línea]. Disponible en: http://www.doubango.org

[25] García Pérez P. Diseño e implementación de la arquitectura de red y seguridad de un sistema de comunicaciones unificadas basado en el protocolo SIP en un entorno empresarial. Proyecto, ETSIT. Universidad Politécnica de Madrid. 2013.

[26] Gonçalves F. Asterisk PBX: Guía de la Configuración. [En Línea]. Disponible en: http://www.asteriskguide.com/

[27] Vásquez, L. Diseño e implementación de un modelo de Telemedicina y Tele-salud rural en la red del MSP de la provincia de Zamora Chinchipe, durante el 2013. Grupo de Bioingeniería y Telemedicina, ETSIT UPM.

9. ANEXOS

MANUAL DE USUARIO

1. INTRODUCCIÓN

En la presente guía se usuario se contempla detalladamente la aplicación y manejo correcto del módulo de videoconferencia de la plataforma de Telemedicina Rural **"Telesalud UTPL TUTUPALY"** con fin de brindar al usuario una ayuda adecuada en su utilización y obtener el mayor rendimiento óptimo y efectividad posibles.

Los requerimientos necesarios para utilizar dicho módulo son los siguientes:

- ✓ Conexión a internet.
- ✓ ***Es altamente recomendable utilizar navegadores Mozilla Firefox o Google Chrome***.
- ✓ Disponer de micrófono, cámara y altavoces.
- ✓ Ser parte del sistema o usuario de la plataforma **"Telesalud UTPL TUTUPALY"**, indistintamente de su rol.
- ✓ Ingreso a la plataforma de Telemedicina con sus credenciales.

El módulo de videoconferencia se encuentra alojada dentro de la plataforma del proyecto antes mencionado, específicamente en la sección o pestaña ***"VIDEOCONFERENCIA".***

El módulo de videoconferencia, permite establecer videoconferencias y ofrecerá distintas funcionalidades de acuerdo con el rol al que el usuario pertenezca, estos son: médico rural, médico especialista y médicos directivos/administrativos.

Ambos roles podrán ver las consultas y respuestas que se han generado así como también las que están pendientes. Se podrán visualizar en el navegador web de su preferencia y además podrán imprimir lo que sea necesario.

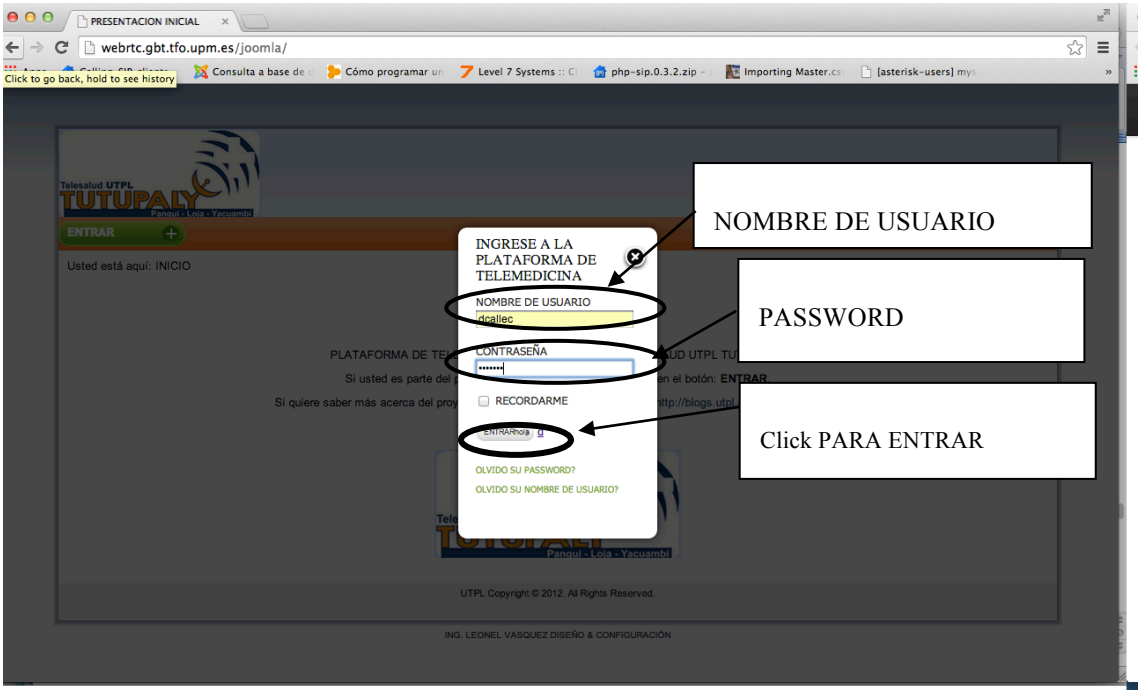
2. ACCESO AL MÓDULO DE VIDECONFERENCIA:

Para acceder a la plataforma en general se utilizará la siguiente dirección web:

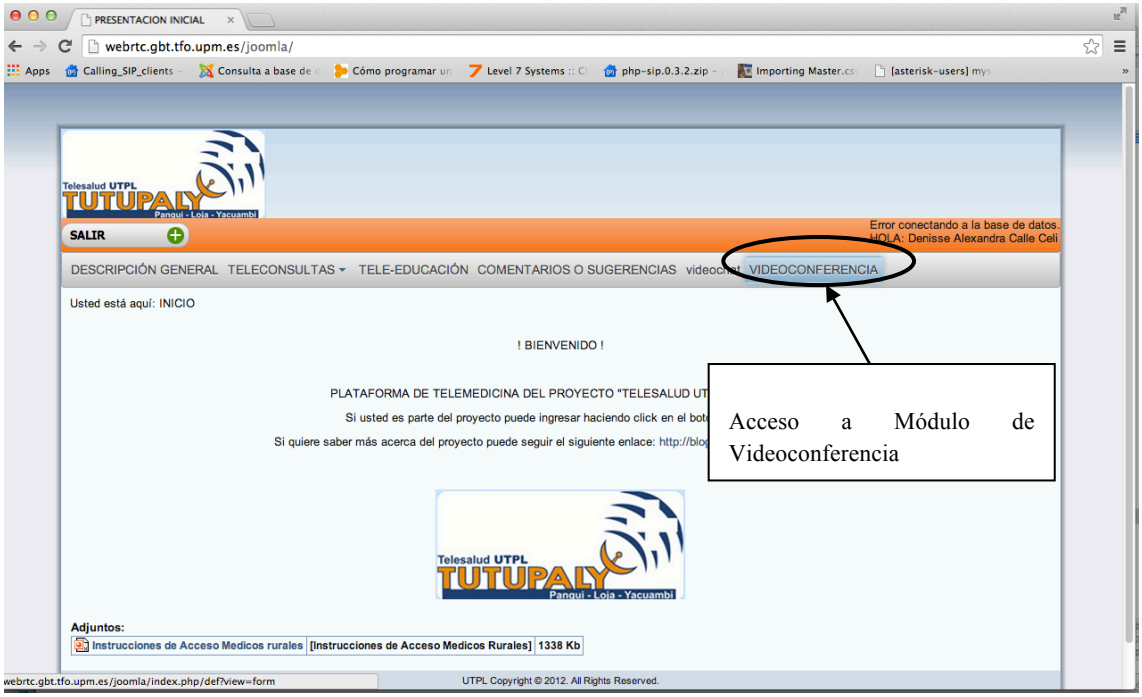
http://webrtc.gbt.tfo.upm.es/joomla/

Es necesario aclarar que esta es una dirección de pruebas y evaluación únicamente, por lo que es posible que en un futuro inmediato sea diferente, no obstante los módulos tendrán las mismas funcionalidades.

La pantalla principal permite el acceso al sistema a toda persona que disponga de nombre de usuario y contraseña, haciendo clic en el botón de ENTRAR []. Ejemplo, usuario: "dcallec", password: ******. Para cuestiones de prueba se ha utilizado el mismo usuario como password.

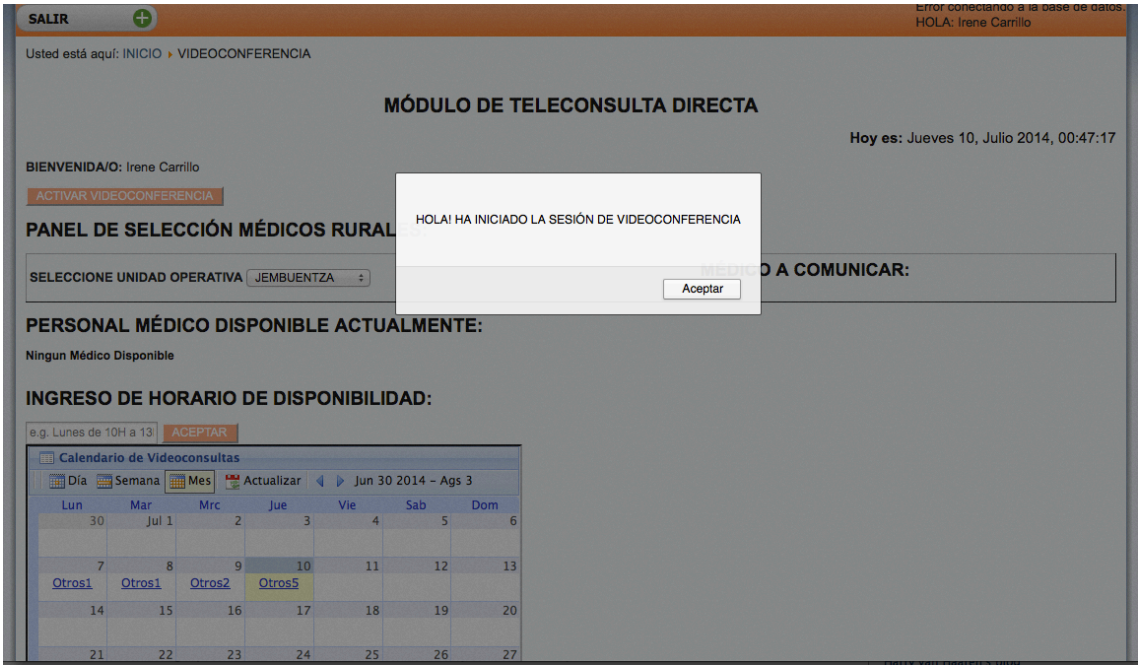


Una vez ingresado al sistema de Telemedicina, seleccionamos en el menú principal la pestaña correspondiente a ***VIDEOCONFERENCIA***



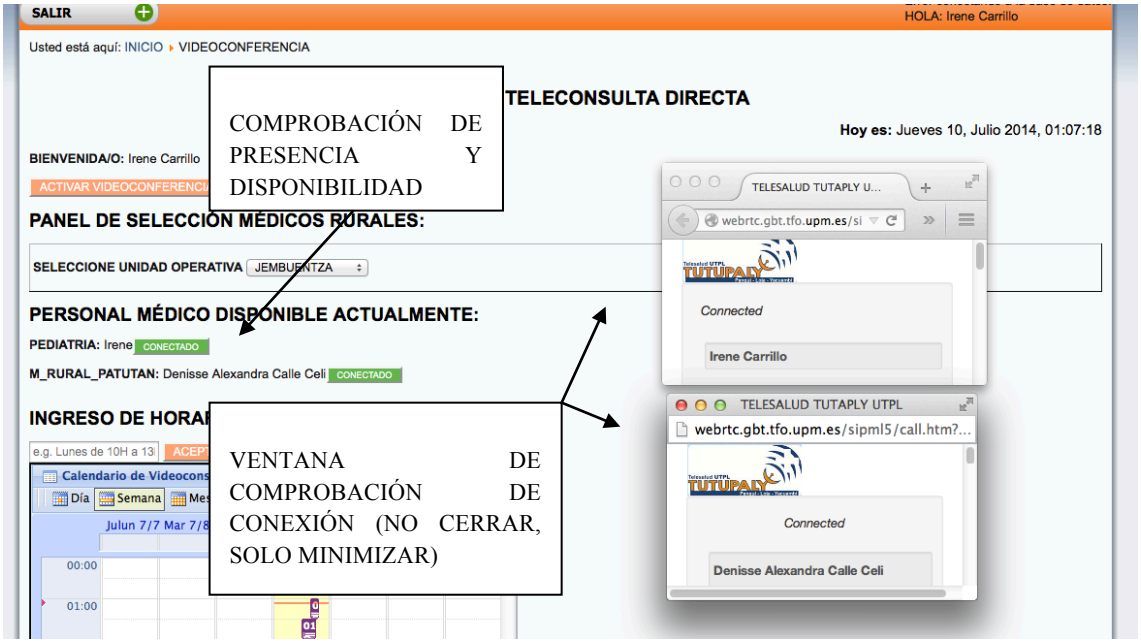
Dentro del menú VIDEOCONFERENCIA, se tendrá la siguiente interfaz si el usuario es un médico rural o especialista, respectivamente.

Para *activar la funcionalidad de videoconferencia* y presentar su disponibilidad en el sistema al resto de usuarios, se tiene que hacer click en el botón **"ACTIVAR VIDEOCONFERENCIA".** Obtendrá la siguiente información de confirmación de inicio de sesión de videoconferencia.


SALIR
HOLA: Irene Carrillo
Usted está aquí: INICIO ▸ VIDEOCONFERENCIA
MÓDULO DE TELECONSULTA DIRECTA
Hoy es: Jueves 10, Julio 2014, 00:47:17
BIENVENIDA/O: Irene Carrillo
ACTIVAR VIDEOCONFERENCIA
PANEL DE SELECCIÓN MÉDICOS RURAL
HOLA! HA INICIADO LA SESIÓN DE VIDEOCONFERENCIA
Aceptar
SELECCIONE UNIDAD OPERATIVA JEMBUENTZA
A COMUNICAR:
PERSONAL MÉDICO DISPONIBLE ACTUALMENTE:
Ningun Médico Disponible
INGRESO DE HORARIO DE DISPONIBILIDAD:
e.g. Lunes de 10H a 13
ACEPTAR
Calendario de Videoconsultas
Día Semana Mes Actualizar Jun 30 2014 – Ags 3
Lun Mar Mrc Jue Vie Sab Dom
Otros1 Otros1 Otros2 Otros5

Si quisiere desactivar la videoconferencia, con lo cual desaparecerá su presencia del sistema de videonferencia y conservará su sesión en la plataforma general; simplemente tiene que cerrar la ventana inicial y le presentará el siguiente mensaje.

Al activar la videoconferencia, seguido al mensaje de inicio tendrá una pequeña ventana en la que le comunicará si usted está conectado o no. ***Es importante no cerrar esta ventana puesto que es la ventana que permitirá comprobar su presencia y disponibilidad para realizar o recibir videollamadas.***

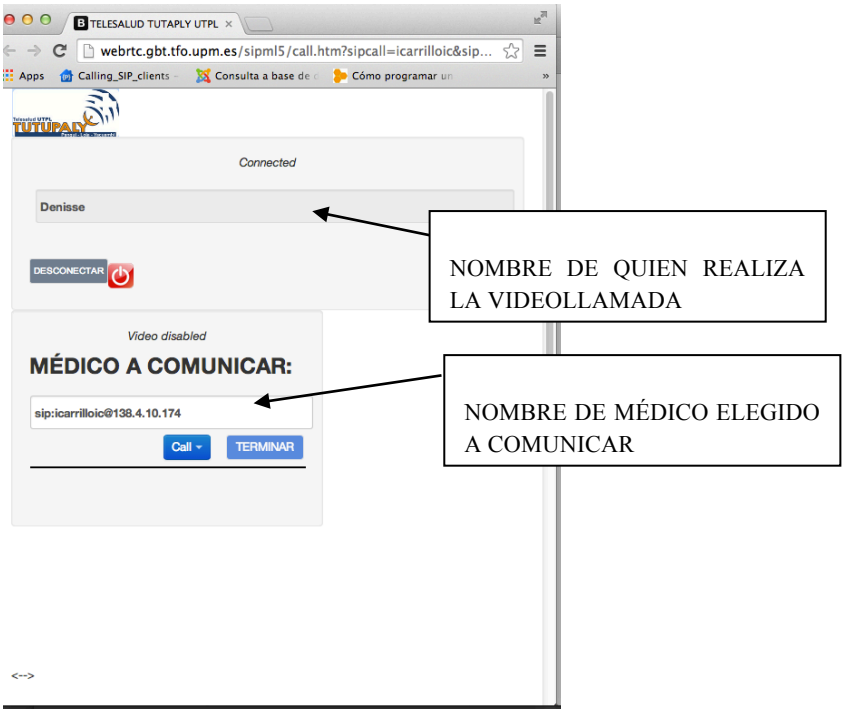


Una vez que vea que está conectado, podrá empezar a realizar videollamadas, así como también recibirlas. Ejemplo: Denisse que es médico rural tratará de establecer comunicación con Irene Carrillo que es especialista en Pediatría que actualmente se encuentra disponible, por tanto ella tiene la posibilidad de seleccionar desde el PANEL DE SELECCIÓN DE MÉDICO ESPECIALISTAS o a su vez directamente desde donde se indica el PERSONAL MÉDICO DISPONIBLE ACTUALMENTE.

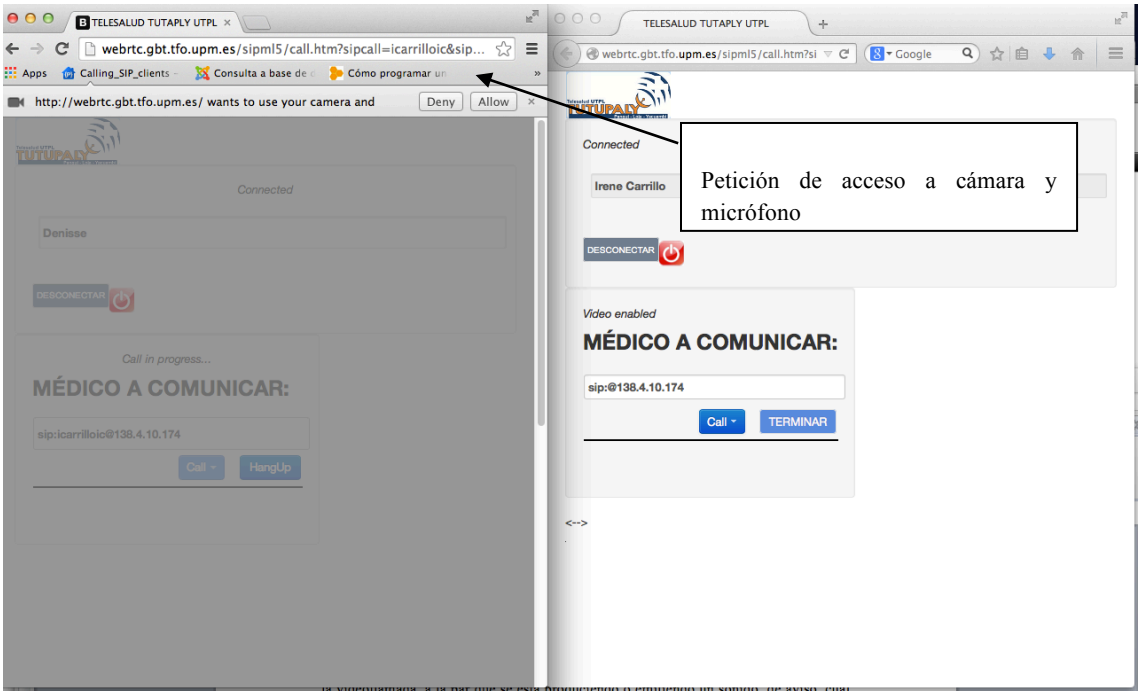
En el caso contrario, de igual manera, los especialistas pueden seleccionar a los médicos rurales desde el PANEL DE SELECCIÓN DE MÉDICO RURAL de acuerdo con la unidad operativa a la que estos pertenecen; o a su vez desde PERSONAL MÉDICO DISPONIBLE ACTUALMENTE.

VIDEOCONFERENCIA
webrtc.gbt.tfo.upm.es/joomla/index.php/defesp?view=form
SALIR
HOLA: Irene Carrillo
Usted está aquí: INICIO ▸ VIDEOCONFERENCIA
MÓDULO DE TELECONSULTA DIRECTA
Hoy es: Jueves 10, Julio 2014, 10:56:38
BIENVENIDA/O: Irene Carrillo
ACTIVAR VIDEOCONFERENCIA
PANEL DE SELECCIÓN MÉDICOS RURALES:
SELECCIONE UNIDAD OPERATIVA PATUTAN
MÉDICO A COMUNICAR:
Denisse Alexandra Calle Celi CONECTADO
Médicos Disponibles otras U.O.:
PERSONAL MÉDICO DISPONIBLE ACTUALMENTE:
PEDIATRIA: Irene CONECTADO
M_RURAL_PATUTAN: Denisse Alexandra Calle Celi CONECTADO
INGRESO DE HORARIO DE DISPONIBILIDAD:
e.g. Lunes de 10H a 13
ACEPTAR
Calendario de Videoconsultas
Día Semana Mes Actualizar Cargando
Julun 7/7 Mar 7/8 Mrc 7/9 Jue 7/10 Vie 7/11 Sab 7/12 Dom 7/13
SELECCIÓN DE MÉDICOS RURALES

Cuando se ha seleccionado el médico a comunicar, aparecerá la siguiente ventana indicando su nombre y la dirección a quien llamará, de esta forma únicamente hace falta dar click en el botón **CALL** para empezar a realizar la videollamada.

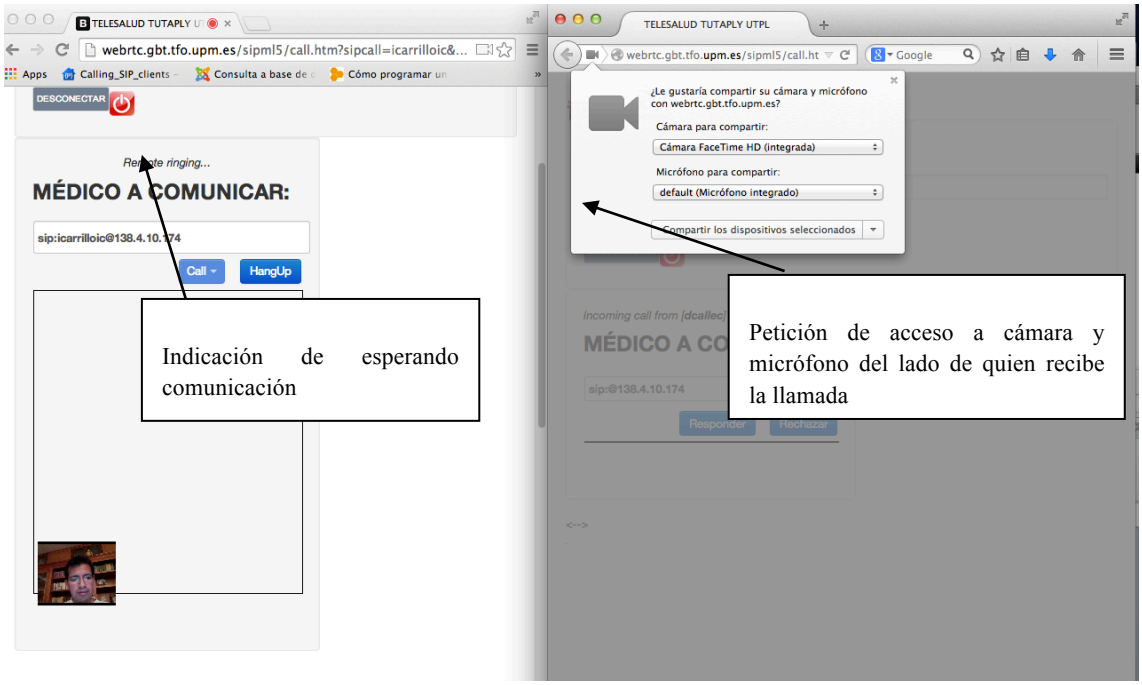


Al presionar el botón de *llamar o Call,* la página le solicitará permiso para acceder a la cámara y micrófono de su computador.

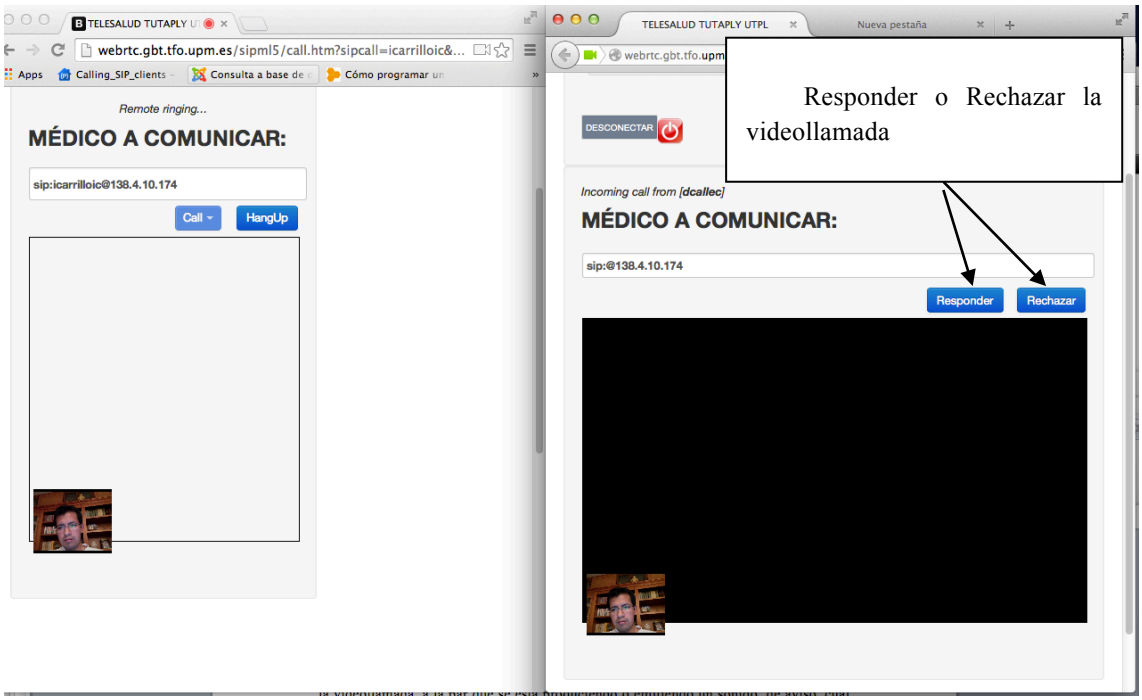


Cuando se ha dado permiso de acceso a cámara y micrófono, se indica un mensaje de comunicando, lo cual en el otro lado, es decir en la persona que está por recibir la videollamada también se le va a solicitar el permiso para acceder a su respectiva cámara y micrófono. Esto a la par que se realiza, se están también

manifestando sonidos de aviso, tanto para el que llama como para el que recibe la llamada.



Concedido el acceso a la cámara y micrófono del lado de quien recibe la llamada, también se le da la oportunidad de responder o rechazar la llamada.

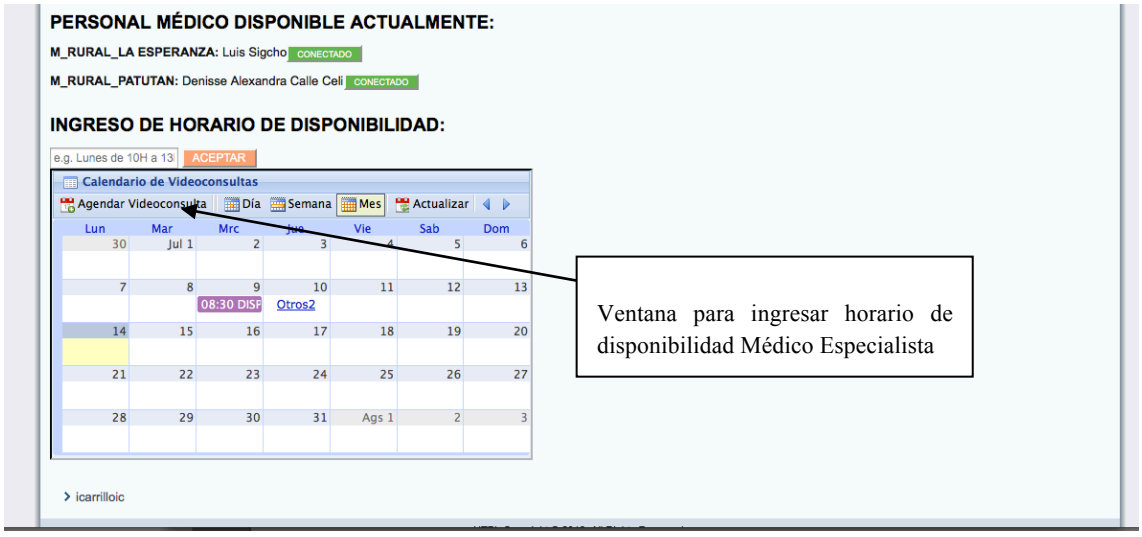


Si responde la llamada se tendrá el establecimiento de la videoconferencia, FELICIDADES!!.

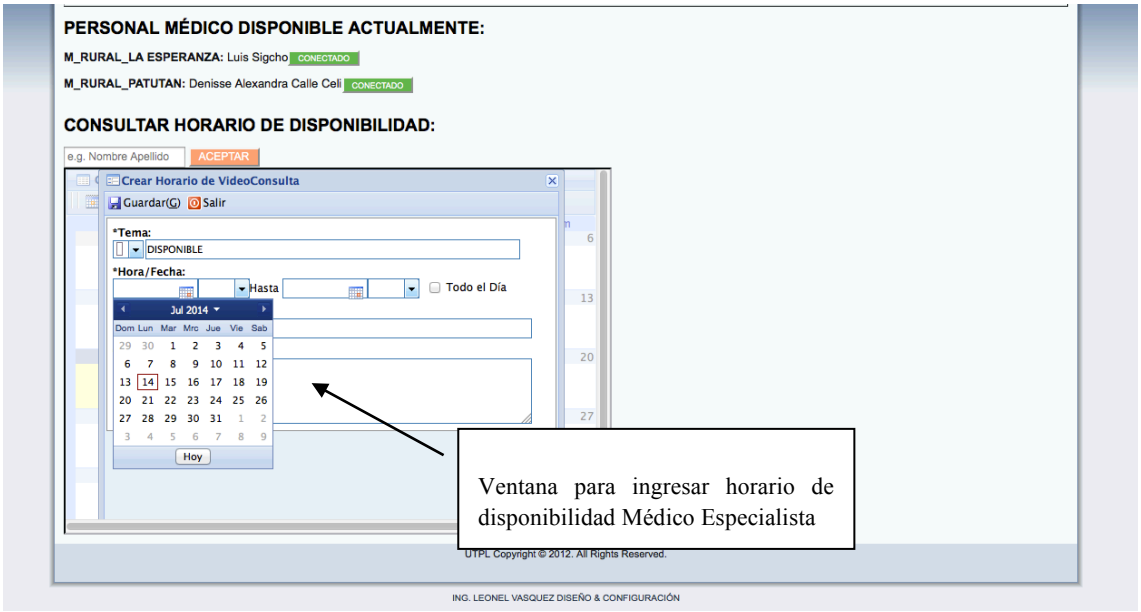
Médico que **recibe** la videoconsulta
Médico que **realiza** la videoconsulta
Médico que **realiza** la videollamada
Médico que **recibe** la videoconsulta

CONSULTA DE DISPONIBILIDAD EN EL CALENDARIO:

En la interfaz de médico especialista y médico rural, dispone de un calendario en el que se podrá consultar los horarios de disponibilidad para videoconferencias de parte de los médicos especialistas.



Aspecto de Calendario de Médico Especialista.



Ingreso de Horario de disponibilidad de Calendario de Médico Especialista.

Aspecto de Calendario de Médico Rural para consulta de horario de médicos Especialistas.

ANEXO 2:

Guía Técnica de Instalación de Webrtc2sip (V 2.5 -2013-10)

La guía técnica que se adjunta es tomada directamente desde el portal web de Doubango Telecom [24].

webrtc2sip - Smart SIP and Media Gateway for WebRTC endpoints

Technical Guide

by

Mamadou DIOP
diopmamadou {AT} doubango[DOT]org

License

webrtc2sip - Smart SIP and Media Gateway for WebRTC endpoints version **2.6.0**
Copyright © 2012-2013 Doubango Telecom <http://www.doubango.org>

webrtc2sip is a free software: you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation, either version 3 of the License, or (at your option) any later version.

webrtc2sip is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the **GNU General Public Licence** along with *webrtc2sip*. If not, see <http://www.gnu.org/licenses/>.

Versioning

Date	Version	SVN revision	Authors	Comments
December 2, 2012	2.0.0	9	Mamadou DIOP	Initial version
January 7, 2013	2.1.0	WEBRTC2SIP: 38+ DOUBANGO: 804+	Mamadou DIOP	1. Add support for DTLS-SRTP (rfc5763 and rfc 5764) 2. Add new command line arguments: ***--config***, ***--help*** and ***--version*** 3. Add new xml configuration entries: ***video-size-pref***, ***enable-rtp-symetric*** *and* ***srtp-type*** 4. Add **verify** option to xml configuration entry **<ssl-certificates />** to allow remote certificates verification. 5. Fix issues: 35, 36, 37, 39, 41, 42 and 43.
January 14, 2013	2.2.0	WEBRTC2SIP: 44+ DOUBANGO: 808+	Mamadou DIOP	1. Adds support for Firefox Nightly 2. Fix issues: 47, 48
March 11, 2013	2.3.0	WEBRTC2SIP: 53+ DOUBANGO: 838+	Mamadou DIOP	1. Adds click-to-call service (http://click2dial.org) 2. Fix issues: 58, 59 and 60
March 26, 2013	2.4.0	WEBRTC2SIP: 64+ DOUBANGO: 856+	Mamadou DIOP	1. Adds support for DTMF relaying 2. Adds support for TCP/TLS outbound 3. Fix issues: 64, 66, 70 and 71
May 06, 2013	2.5.0	WEBRTC2SIP:86+ DOUBANGO: 884+	Mamadou DIOP	1. Adds support for OPUS audio codec 2. Fix issues: 13, 26, 77, 78, 81, 85, 88
June 03, 2013	2.5.1	WEBRTC2SIP: 90+ DOUBANGO: 895+	Mamadou DIOP	1. Add new xml configuration entries: ***stun-server*** and ***enable-icestun***. 2. Fix issues: 62, 92 and 95
October 07, 2013	2.6.0	WEBRTC2SIP: 116+ DOUBANGO: 1002+	Mamadou DIOP	1. Make the server more robust to DDoS attacks 2. Add new xml configuration entries: ***max-fds***

Table of Contents

1 Foreword....5
2 Scope....5
3 Architecture....5
3.1 SIP Proxy module....5
3.2 RTCWeb Breaker....7
3.3 Media Coder....9
3.4 Click-to-Call....9
3.4.1 SMTP client....10
3.4.2 HTTPS server....10
3.4.3 Database connector....10
3.4.4 JSON API....10
4 Configuration....10
5 Building source code....16
5.1 Building Doubango IMS Framework....17
5.2 Building webrtc2sip....20
5.3 Running webrtc2sip....21
5.3.1 Command line arguments....21
6 Testing the gateway....21
7 Interoperability....21
7.1 Servers....21
7.1.1 Asterisk....21
7.1.2 FreeSWITCH....22
7.2 Web Browsers....22
7.2.1 Google Chrome....22
7.2.2 Firefox Nightly....22
7.2.3 Firefox, Safari, IE and Opera....22
7.2.4 Ericsson Bowser....23
7.3 JavaScript SIP stacks....23
8 Security issues....24

Table of Figures

Figure 1: Architecture....5
Figure 2: SIP Proxy architecture....5
Figure 3: RTCWeb Breaker architecture....7
Figure 4: Enabling RTCWeb Breaker on sipml5....7
Figure 5: Media Coder architecture....9
Figure 6: click-to-call components....10

Table of Samples

Sample 1: config.xml....11

1 Foreword

RTCWeb (a.k.a *WebRTC*) stands for **R**eal-**T**ime **C**ommunication and is a new technology being drafted by the **W**orld **W**ide **W**eb **C**onsortium (W3C) and **IETF** groups. This technology has the ambition to bring native real-time features (audio, video and arbitrary data) to the web browsers without requiring additional plugins.

SIP stands for **S**ession **I**nitiation **P**rotocol and is a signaling protocol defined by the IEFT in *RFC 3261*. *SIP* is widely used today to manage VoIP (**V**oice **o**ver **IP**) communication sessions and has been chosen as signaling protocol for **N**ext **G**enerations **N**etworks such as *IMS* (**IP M**ultimedia **S**ubsystem) or *LTE* (**L**ong **T**erm **E**volution). The protocol has quickly become the de facto standard used to interconnect the IP world (Internet) with the PSTN (circuit-switched telephone networks).

webrtc2sip is a smart and powerful gateway using *RTCWeb* and *SIP* to turn your browser into a phone with audio, video and SMS capabilities. The gateway allows your web browser to make and receive calls from/to any SIP-legacy network or PSTN. As an example, you will be able to make a call from your preferred web browser to a mobile or fixed phone.

2 Scope

This technical guide is a reference document explaining why you need *webrtc2sip* and how to leverage its power.

3 Architecture

The gateway contains four modules: *SIP Proxy*, *RTCWeb Breaker*, *Media Coder and click-to-call service*.

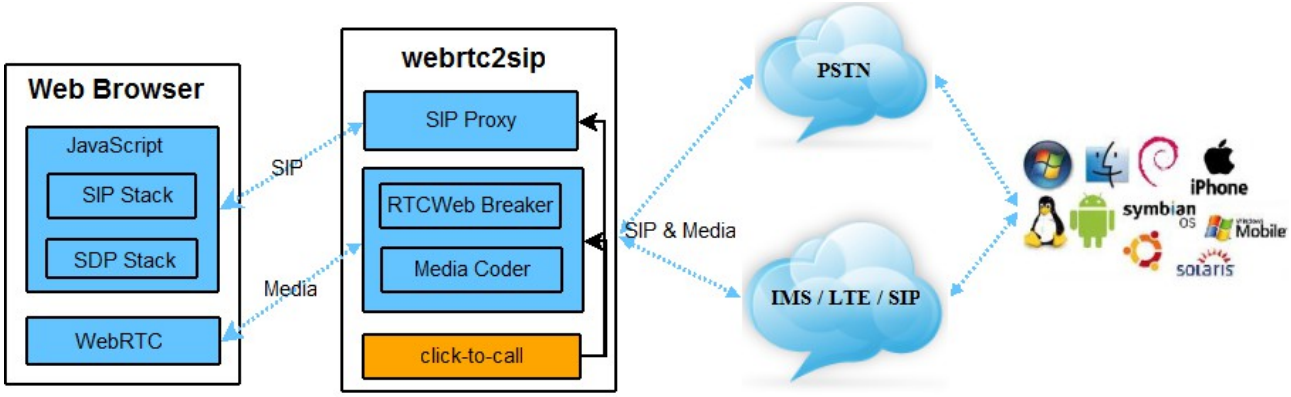


Figure 1: Architecture

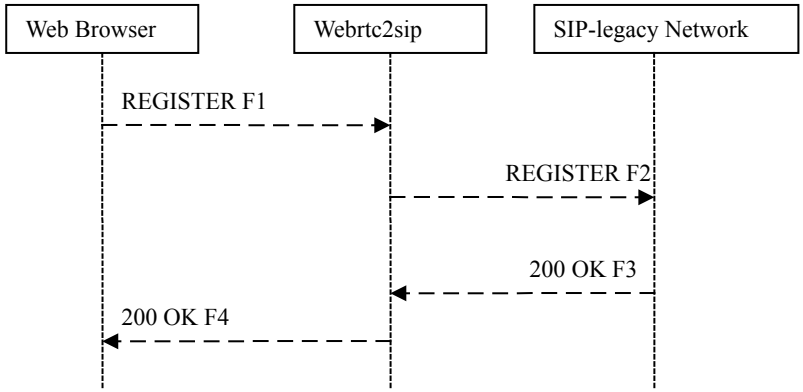
The HTML SIP client is any endpoint implementing draft-ibc-sipcore-sip-websocket-06. We highly recommend using sipML5 which is known to work and provide good performances.

3.1 SIP Proxy module

Figure 2: SIP Proxy architecture

The role of the *SIP Proxy* module is to convert the SIP transport from WebSocket protocol to UDP, TCP or TLS which are supported by all SIP-legacy networks. If your provider or hosted server supports SIP over WebSocket (e.g. Asterisk or Kamailio) then, you can bypass the module and connect the client directly to the endpoint. Bypassing the *SIP Proxy* is not recommended if you're planning to use the *RTCWeb Breaker* or *Media Coder* modules as this will requires maintaining two different connections.

There are no special requirements for the end server to be able to talk to the *Proxy module*.




```
F1 REGISTER Web Browser -> webrtc2sip (transport WS)
REGISTER sip:proxy.example.com SIP/2.0
Via: SIP/2.0/WS df7jal23ls0d.invalid;branch=z9hG4b5
From: sip:browser@example.com;tag=abc
To: sip:browser@example.com
Call-ID: abcdefghijklmnopqrstuvwxyz
CSeq: 1 REGISTER
Max-Forwards: 70
Contact: <sip:browser@df7jal23ls0d.invalid;transport=ws>
```

This request contains an invalid IP address in the `Contact` (**df7jal23ls0d.invalid**) and `Via` headers because there is no way for the browser to retrieve its local binding *IP:Port address*. The transport type is WebSocket (**ws**). A SIP-legacy server cannot handle this request as the transport is probably not supported and the IP address and port are not valid (not reachable), this is why we need the *SIP Proxy* module to patch the request before forwarding.

```
F2 REGISTER webrtc2sip -> SIP-legacy Network (transport UDP)
REGISTER sip:proxy.example.com SIP/2.0
Via: SIP/2.0/UDP 66.66.66.66:5060;branch=z9hG4b5;rport
Via: SIP/2.0/TCP 192.168.0.9:55210;rport;branch=z9hG4b6;ws-hacked=WS
From: sip:browser@example.com;tag=abc
To: sip:browser@example.com
Call-ID: abcdefghijklmnopqrstuvwxyz
CSeq: 1 REGISTER
Max-Forwards: 70
Contact: <sip:browser@66.66.66.66:5060;transport=udp>
```

The `Via` header is patched to use a well-known protocol (*TCP*) and to use the IP address and port (**192.168.0.9:55210**) from which the request has been received (WebSocket connection).

The *SIP Proxy* adds it's own `Via` header (**66.66.66.66:5060**) where it's willing to receive the

response. The same address is used in the Contact header for incoming requests (e.g. `INVITE`).

Before forwarding the request the SIP Proxy determines the destination address using the following algorithm:

```
char* dst_host = get_host(request_uri); // dst_host = "proxy.example.com"
int dst_port = 5060;
if(has_route(request)){ // there a route header
      dst_host = get_host(first_route);
      dst_port = get_port(first_route);
}
if((dns_result = dns_srv(dns_naptr(dst_host)))){
      dst_host = get_host(dns_result);
      dst_port = get_port(dns_result);
}
```

3.2 RTCWeb Breaker

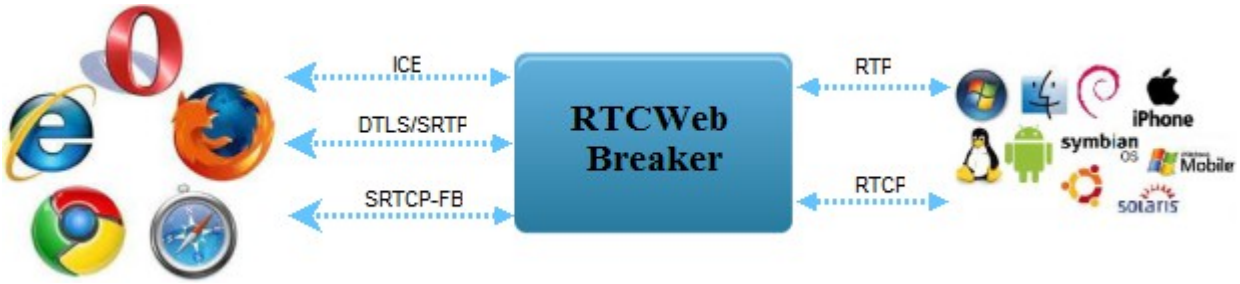


Figure 3: RTCWeb Breaker architecture

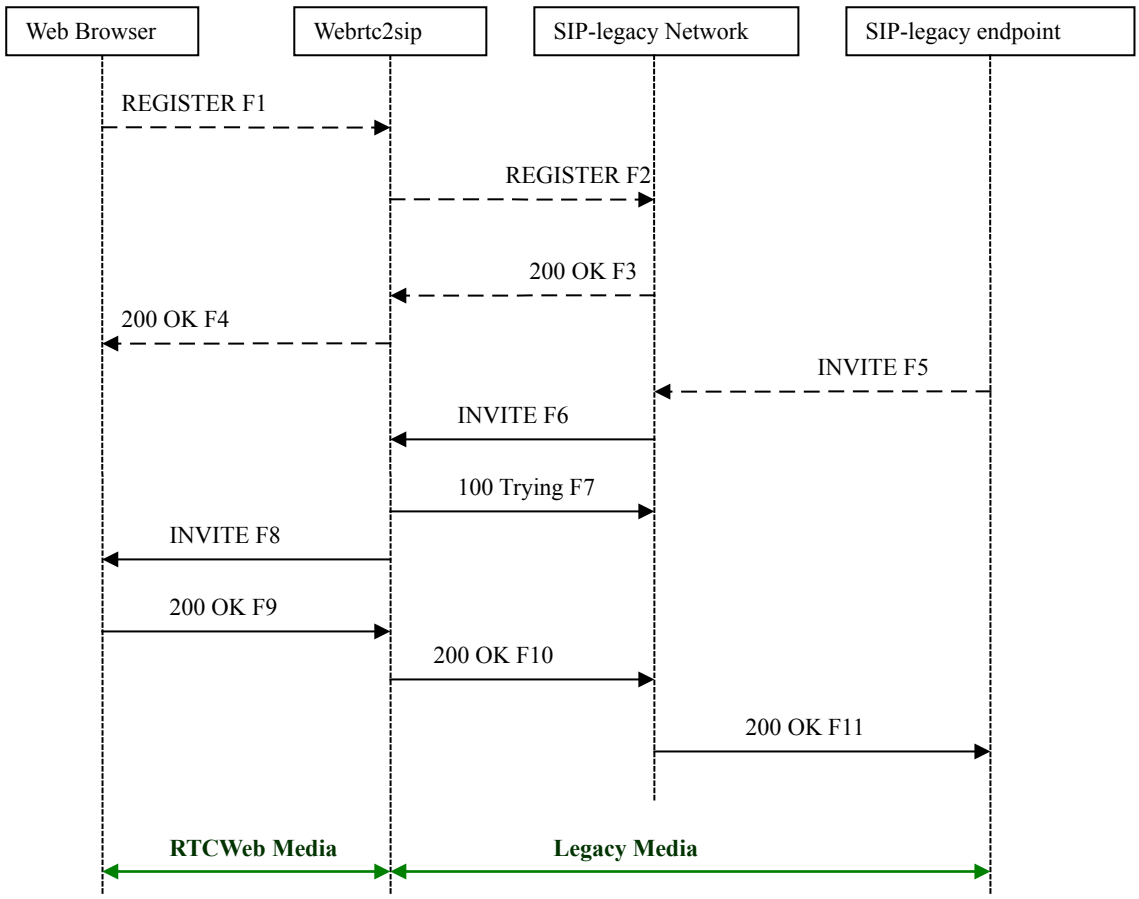
The RTCWeb specifications make support for **ICE** and **DTLS/SRTP** mandatory. The problem is that many SIP-legacy endpoints (e.g. PSTN network) do not support these features. It's up to the *RTCWeb Breaker* to negotiate and convert the media stream to allow these two worlds to interop.

For example, FreeSWITCH do not support ICE which means it requires the *RTCWeb Breaker* in order to be able to connect the browser to a SIP-legacy endpoint.

The *RTCWeb Breaker* is disabled by default and it's up to the client to enable it before registering to the server.

To activate the *RTCWeb Breaker*, the client must include "*rtcweb-breaker=yes*" as Uri parameter of its AoR (**A**ddress **o**f **R**ecord). When the module is enabled it acts as a *b2bua* (**b**ack **2** **b**ack **u**ser **a**gent) by answering to the *INVITE* and making a new one.

Figure 4: Enabling RTCWeb Breaker on sipml5



F1 REGISTER web browser -> webrtc2sip (transport WSS)

-- TODO--

F2 REGISTER webrtc2sip -> SIP-legacy network (transport UDP)

-- TODO --

F3 200 OK SIP-legacy network -> webrtc2sip (transport UDP)

--TODO--

F4 200 OK webrtc2sip -> web browser(transport WSS)

--TODO--

F4 200 OK webrtc2sip -> web browser(transport WSS)

--TODO--

F5 INVITE SIP-legacy endpoint -> SIP-legacy network (transport UDP)

--TODO--

F6 INVITE SIP-legacy network -> webrtc2sip (transport UDP)

--TODO--

F7 100Trying webrtc2sip -> SIP-legacy network (transport UDP)

--TODO--

F8 INVITE webrtc2sip -> web browser (transport WSS)

--TODO--

```
F9 200 OK web browser -> webrtc2sip (transport WSS)
```

--TODO--

```
F10 200 OK webrtc2sip -> SIP-legacy network (transport UDP)
```

--TODO--

```
F11 200 OK SIP-legacy network -> SIP-legacy endpoint (transport UDP)
```

--TODO--

3.3 Media Coder

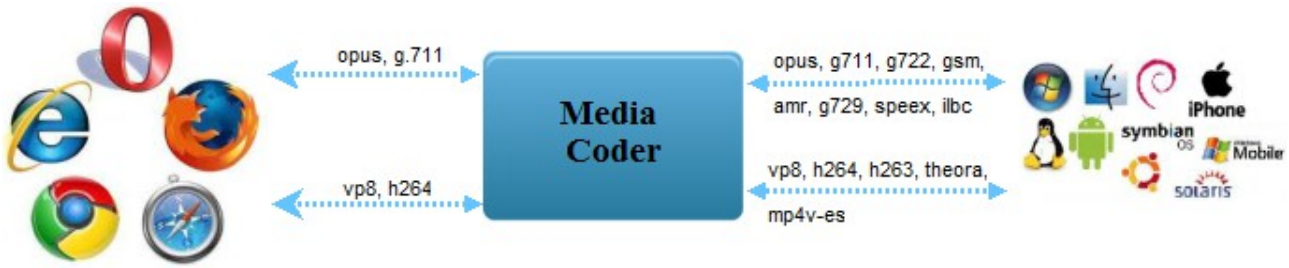


Figure 5: Media Coder architecture

The RTCWeb standard defined two MTI (**M**andatory **T**o **I**mplement) audio codecs: *opus* and *g.711*.

For now there are intense discussions about the MTI video codecs. The choice is between *VP8* and *H.264*. VP8 is royalty-free but not widely deployed while H.264 *AVC* is not free but widely deployed. Google has decided to use *VP8* in Chrome while Ericsson uses *H.264 AVC* in Bowser. Mozilla and Opera Software will probably use *VP8* and Microsoft *H.264 AVC*. As an example, the *Media Coder* will allow to make video calls between Chrome and Bowser. Another example is calling a Telepresence system (e.g. Cisco) which most likely uses *H.264 SVC* from Chrome.

The *Media Coder* is enabled using the xml configuration file and requires *RTCWeb breaker* module to be enabled.

3.4 Click-to-Call

This is more a service than a module as it's a complete SIP click-to-call solution based on the three other components. The goal is to allow any person receiving your mails, visiting your website, reading your twitts, watching your Facebook/Google+ profile to call you on your mobile phone with a single click.

The client is hosted at http://click2dial.org/

A short user guide is available at http://click2dial.org/u/ug.htm

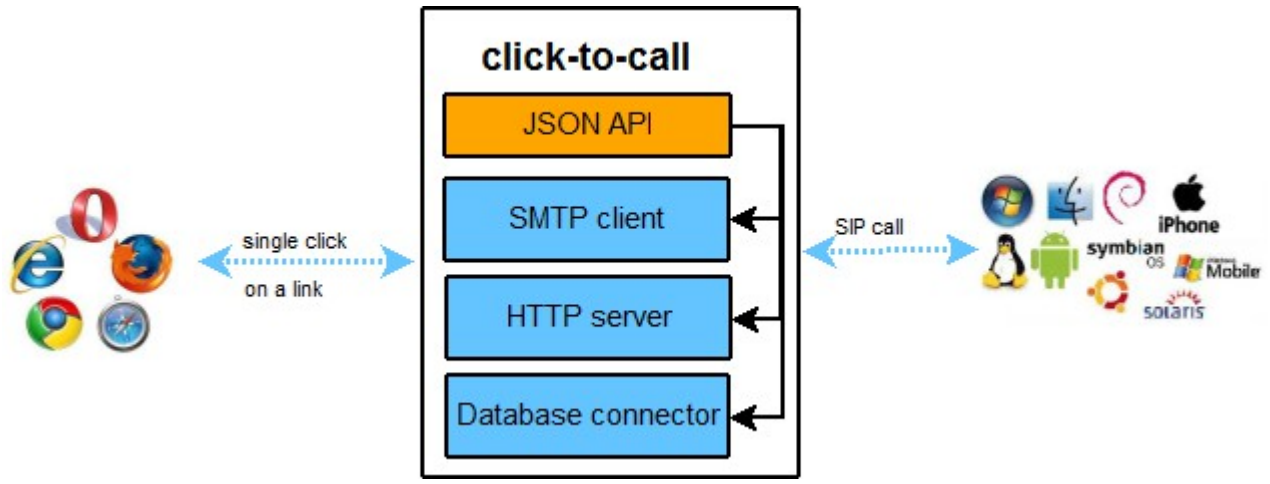


Figure 6: click-to-call components

3.4.1 SMTP client

This component is used to send activation mails for newly registered users. It's coded from scratch and has no external dependencies.

3.4.2 HTTPS server

For now, the HTTPS server is used exclusively by the JSON API to exchange content between the browser and the click-to-call service. It's coded from scratch and depends on tinyHTTP (from Doubango VoIP framework).

3.4.3 Database connector

Agnostic API functions to connect to any database used to store users information, configuration…

In this beta version, only SQLite is supported. Next release will add support to MySQL and SQL Server.

3.4.4 JSON API

The JSON API is used to authenticate the users and manage their accounts. The documentation will be released soon at http://click2dial.org/doc.htm. On the server-side, the parser is based on Json-Cpp.

4 Configuration

The gateway is configured using an xml file named *config.xml* and stored in the same folder where the gateway is running.

```
<?xml version="1.0" encoding="utf-8" ?>
<config>
  <debug-level>INFO</debug-level>

  <transport>udp;*;10060</transport>
  <transport>ws;*;10060</transport>
  <transport>wss;*;10062</transport>

  <enable-rtp-symetric>yes</enable-rtp-symetric>
  <enable-100rel>no</enable-100rel>
```

```
  <enable-media-coder>no</enable-media-coder>
  <enable-videojb>yes</enable-videojb>
  <video-size-pref>vga</video-size-pref>
  <rtp-buffsize>65535</rtp-buffsize>
  <avpf-tail-length>100;400</avpf-tail-length>
  <srtp-mode>optional</srtp-mode>
  <srtp-type>sdes;dtls</srtp-type>
  <dtmf-type>rfc4733</dtmf-type>

  <codecs>opus;pcma;pcmu;gsm;vp8;h264-bp;h264-mp;h263;h263+</codecs>
  <codec-opus-maxrates>48000;48000</ codec-opus-maxrates >

  <stun-server>stun.l.google.com;19302;stun-user;stun-password</stun-server>
  <enable-icestun>yes</enable-icestun>

  <max-fds>65535</max-fds>

  <nameserver>66.66.66.66</nameserver>
  <nameserver>77.77.77.77</nameserver>

  <ssl-certificates>
       /tmp/priv.pem;
       /tmp/pub.pem;
       /tmp/cacert.pem;
       no
</ssl-certificates>

<!-- ***CLICK-TO-CALL SERVICE*** -->
  <transport>c2c;*;10070</transport>
  <transport>c2cs;*;10072</transport>
  <database>sqlite;*</database>
  <account-mail>smtps;*;*;e.org;465;noreply@e.org;noreply@e.org;mysecret </account-
mail>
  <account-sip-caller>*;sip:13131313@b.c;13131313;b.c;mysecret</account-sip-caller>
  <account-sip-caller>*;sip:13131313@a.c;13131313;a.c;mysecret</account-sip-caller>


</config>
```

Sample 1: config.xml

<debug-level />

Define the minimum debug-level to display.

Format: debug-level-value

Debug-level-value = INFO | WARN | ERROR | FATAL

<transport />

Each entry defines a protocol, local IP address and port to bind to.

Format: proto-value;local-ip-value;local-port-value

proto-value: udp | tcp | tls | ws | wss | c2c | c2cs

"ws" protocol defines WebSocket and "wss" the secure version. At least one WebSocket transport must be added to allow the web browser to connect to the server. The other protocols (tcp, tls and udp) are used to forward the request from the web browser to the SIP-legacy network. "C2c" and "c2cs" are used for the click-to-call service and runs on top of HTTP or HTTPS protocols respectively.

local-ip-value: Any valid IP address. Use star (*) to let the server choose the best local IP address to bind to. Examples: udp;*;5060 or ws;*;5061 or wss;192.168.0.10;5062

local-port-value: Any local free port to bind to. Use star (*) to let the server choose the best free port to bind to. Examples: udp;*;*, ws;*;* or wss;*;5062

<enable-rtp-symetric />

Format: enable-rtp-symetric-value

enable-rtp-symetric-value: yes | no

Available since: 2.1.0

This option is used to force symmetric RTP and RTCP streams to help NAT and firewall traversal. It only applies on remote RTP/RTCP as local stream is always symmetric. If both parties (remote and local) have successfully negotiated ICE candidates then, none will be forced to use symmetric RTP/RTCP.

An RTP/RTCP stream is symmetric if the same port is used to send and receive packets. This helps for NAT and firewall traversal as the outgoing packets open a pinhole for the ongoing ones.

Let's imagine you have a server on public network and a client on private network:

1. Server: Public IP address is **1.1.1.1**
2. Client: Private IP address is **2.2.2.2** and public IP address is **1.1.1.2**
3. The SDP from the client to the sever will contain client's private IP address (**2.2.2.2**) which is not reachable
4. The RTP/RTCP packets from the client to the server will be received with source IP address equal to the client's public IP address (**1.1.1.2**)
5. If <enable-rtp-symetric /> option is used then, the server will send RTP/RTCP packets to **1.1.1.2** (learnt from the received packets) instead of **2.2.2.2** which is private.

<enable-100rel>

Format: enable-100rel-value

enable-100rel-value: yes|no

Indicates whether to enable SIP 100rel extension.

<enable-media-coder />

Format: enable-media-coder-value

enable-media-coder-value: yes|no

Indicates whether to enable the Media Coder module or not. This option requires the RTCWeb Breaker to be enabled at the web browser level. When the Media Coder is enabled the gateway acts as a b2bua and both audio and video streams are transcoded if the remote peers don't share same codecs.

<enable-videojb />

Format: enable-videojb-value

enable-videojb-value : yes | no

This option is only useful if the RTCWeb Breaker module is enabled at the web browser side. Enabling video jitter buffer gives better quality and improve smoothness. No RTCP-NACK messages will be sent to request dropped RTP packets if this option is disabled.

<video-size-pref />

Format: video-size-pref-value

video-size-pref-value: sqcif | qcif | qvga | cif | hvga | vga | 4cif | svga | 480p | 720p | 16cif | 1080p

Available since: 2.1.0

This option defines the preferred video size to negotiate with the peers. There is no guarantee that the exact size will be used: ***video size to use = Min (Preferred, Proposed);***

<rtp-buffsize />

Format: rtp-buffsize-value

rtp-buffsize-value: Any positive 32 bits integer value. Recommended: 65535.

Code usage:

setsockopt(SOL_SOCKET, SO_RCVBUF, rtp-buffsize-value);

setsockopt(SOL_SOCKET, SO_SNDBUF, rtp-buffsize-value);

Defines the internal buffer size to use for RTP sockets. The higher this value is, the lower will be the RTP packet loss. Please note that the maximum value depends on your system (e.g. 65535 on Windows). A very high value could introduce delay on video stream and it's highly recommended to also enable videojb option.

<avpf-tail-length />

Format: avpf-tail-length-min;avpf-tail-length-max

avpf-tail-length-min: Any positive 32 bits integer

avpf-tail-length-max: Any positive 32 bits integer

Defines the minimum and maximum tail length used to honor RTCP-NACK requests. This option require the Media Breaker module to be enabled on the web browser size. The higher this value is, the better will be the video quality. The default length will be equal to the minimum value and it's up to the server to increase this value depending on the number of unrecoverable packet loss. The final value will be at most equal to the maximum defined in the xml file. Unrecoverable packet loss occurs when the b2bua receives an RTCP-NACK for a sequence number already removed (very common when network RTT is very high or bandwidth very low).

<srtp-mode />

Format: srtp-mode-value

srtp-mode-value: none | optional | mandatory

Defines the SRTP mode to use for negotiation when the RTCWeb Breaker is enabled. Please note that only optional and mandatory modes will work when the call is to a WebRTC endpoint.

Based on the mode, the SDP for the outgoing INVITEs will be formed like this:

none: profile = ***RTP/AVP*** *||| neither crypto lines or certificate fingerprints*

optional: profile = ***RTP/AVP*** *||| two crypto lines if <srtp-type /> includes*

'SDES' plus certificate fingerprints if **<srtp-type />** *include 'DTLS'.*

mandatory: profile = ***RTP/SAVP*** *if* **<srtp-type />** *is equal to 'SDES' or* ***UDP/TLS/RTP/SAVP*** *if* **<srtp-type />** *is equal to 'DTLS' ||| two crypto lines if* **<srtp-type />** *is eaqual to 'SDES' or certificate fingerprints if* **<srtp-type />** *is equal to 'DTLS'*

<srtp-type />

*Format: srtp-type-value; (srtp-type-value)**

srtp-type-value: sdes | dtls

Available since: 2.1.0

Defines the list of all supported SRTP types. Defining multiple values only make sense if the **<srtp-mode />** *value is optional which means we want to negotiate the best one.*

Please note that DTLS-SRTP requires valid TLS certificates and source code must be compiled with OpenSSL version 1.0.1 or later.

<dtmf-type />

Format: dtmf-type-value

dtmf-type-value: rfc4733 | rfc2833

Available since: 2.4.0

Defines the DTMF type to use when relaying the digits. Requires the RTCWeb Breaker to be enabled. ***rfc4733*** *will sends the DTMF digits using RTP packets while* ***rfc2833*** *uses SIP INFO.*

<codecs />

*Format: codec-name (; codec-name)**

codec-name: opus|pcma|pcmu|amr-nb-be|amr-nb-oa|speex-nb|speex-wb|speex-uwb|g729|gsm|g722|ilbc|h264-bp|h264-mp|vp8|h263|h263+|theora|mp4v-es

Defines the list of all supported codecs. Only G.711 and G.722 are natively supported and all other codecs have to be enabled when building the Doubango IMS Framework source code.

Each codec priority is equal to its position in the list. First codecs have highest priority.

<stun-server />

Format: server-fqdn-value; server-port-value; user-name-value; user-password-value

server-fqdn-value: A valid IPv4/v6 address or host name.

server-port: A valid port number.

user-name-value: The login to use for TURN authentication. Use star (*) to ignore.

user-password-value: The password to use for TURN authenetication. Use star (*) to ignore.

Defines the STUN/TURN server to use to gather reflexive addresses for the ICE candidates. If no server is defined then, a default one will be used. The default STUN server is numb.viagenie.ca:3478.

<enable-icestun />

Format: enable-icestun-value

enable-ice-stun-value: yes | no

Defines whether to use STUN to gather reflexive addresses or not. This option is useful when the server is on a public network or all peers are on the same local network.

Disabling STUN for ICE will speed up the call setup.

<codec-opus-maxrates />

Format: maxrate-playback-value; maxrate-capture-value

maxrate-playback-value: 8000|12000|16000|24000|48000

maxrate-capture-value: 8000|12000|16000|24000|48000

Defines the maximum playback and capture rates to negotiate. The final rates to use will be min(offer, answer). Default value = 48000 for both.

The higher this value is, the better will be the voice quality. The bandwidth usage is proportional to the value. In short: high value = high bandwidth usage = good voice quality.

max-fds

Format: max-fds-value

Available since: 2.6.0

max-fds-value: Any integer value from 1 to 65535.

Defines the number of file descrriptors (FDs) the process is allowed to open. The FDs include the pipes and sockets only.

Setting this value is like running ***ulimit -n max-fds-value*** *on Linux.*

<nameserver />

Format: nameserver-value

nameserver-value: Any IPv4 or IPv6 address.

Defines additional entries for DNS servers to use for SRV and NAPTR queries. Please note that this option is optional and should be used carefully.

On Windows and OS X the server will automatically load these values using APIs provided by the OS. On linux, the values come from /etc/resolv.conf. The port must not be defined and the gateway will always use 53.

<ssl-certificates />

Format: private-key-value;public-key-value;cacert-key-value; verify-value

private-key-value: A valid path to a PEM file.

public-key-value: A valid path to a PEM file.

*cacert-key-value: A valid path to a certificate autority file. Should be equal to *.*

Verify-value: Yes | No. This additional option is only available since version 2.1.0. It indicates whether the connection should fail if the remote peer certificates are missing or do not match.This option only applies to TLS/SIP or WSS and is useless for DTLS-SRTP as certificates are required.

Code usage:

```
SSL_CTX_use_PrivateKey_file(ssl_ctx, private-key-value, SSL_FILETYPE_PEM);
SSL_CTX_use_certificate_file(ssl_ctx, public-key-value, SSL_FILETYPE_PEM);
SSL_CTX_load_verify_locations(ssl_ctx, cacert-key-value, CaPath);
```

<database />

Format: db-type-value;db-connection-info-value

Available since: 2.3.0

db-type-value: sqlite | mysql. For now only "sqlite" is supported.

db-connection-info-value: A valid path to the database file if an embeded db is used (e.g. sqlite), otherwise it's an escaped connection string. Use star (*) to let the server use a default value.

For now this configuration entry is only used for the click-to-call service.

<account-mail />

Format: scheme-value;local-ip-value;local-port-value;smtp-host-value;smtp-port-value;email-value;auth-name-value;auth-pwd-value

Available since: 2.3.0

scheme-value: smtp | smtps

local-ip-value: A valid local host name or IP address to be used by the SMTP client. Use star (*) to let the server use the best value.

local-port-value: A valid local port number to be used by the SMTP client. Use star (*) to let the server use a random value.

smtp-host-value: A valid host name or IP address of the SMTP server.

smtp-port-value: A valid port of the SMTP server.

email-value: Email address used as sender.

auth-name-value: Authorization name used to authenticate to the SMTP server. Most probably same value as your email address (email-value).

auth-pwd-value: Password used to authenticate to the SMTP server.

The email account is used to send activation mails to the newly registed users.

<account-sip-caller />

Format: displayname-value;impu-value;impi-value;realm-value;password-value

Available since: 2.3.0

displayname-value: SIP account display name. Optional.

impu-value: Public Identity. Must be a valid SIP address (e.g. sip:003@example.org).

impi-value: Private Identity (a.k.a authorization name) for authentication. Most probably the user part of the Public Identity (e.g. 003).

realm-value: SIP domain name (e.g. example.org). Should be same as the domain name in the Public Identity.

password-value: SIP authentication password.

The SIP account callers are used to make calls to users by the click-to-call service. The callers in the config.xml file are globals (shared by all users) and are override when a user define one using the JSON API.

5 Building source code

This section explains how to build the project using CentOS 64 but could be easily adapted for Linux, Windows or OS X.

webrtc2sip gateway depends on Doubango IMS Framework v2.0.

1. Preparing the system

```
sudo yum update
```

```
sudo yum install make libtool autoconf subversion git cvs wget libogg-devel gcc gcc-c++
pkgconfig
```

5.1 Building Doubango IMS Framework

Doubango is an IMS framework and contains all signaling protocols (SIP, SDP, WebSocket…) and media engine (RTP stack, audio/video codecs…) required by webrtc2sip gateway.

The first step is to checkout Doubango 2.0 source code:

```
svn checkout http://doubango.googlecode.com/svn/branches/2.0/doubango doubango
```

1. Building **libsrtp**

libsrtp is required.

```
git clone https://github.com/cisco/libsrtp/
cd libsrtp
CFLAGS="-fPIC" ./configure --enable-pic && make && make install
```

2. Building **OpenSSL**

OpenSSL is required if you want to use the *RTCWeb Breaker* module or Secure WebSocket transport (WSS). **OpenSSL version 1.0.1 is required if you want support for DTLS-SRTP**.

This section is only required if you don't have OpenSSL installed on your system or using version prior to 1.0.1 and want to enable DTLS-SRTP.

A quick way to have OpenSSL may be installing `openssl-devel` package but this version will most likely be outdated (prior to 1.0.1). Anyway, you can check the version like this: `openssl version`.

```
wget http://www.openssl.org/source/openssl-1.0.1c.tar.gz
tar -xvzf openssl-1.0.1c.tar.gz
cd openssl-1.0.1c
./config shared --prefix=/usr/local --openssldir=/usr/local/openssl && make && make in-
stall
```

3. Building **libspeex** and **libspeexdsp**

libspeex (audio codec) is optional and libspeexdsp (audio processing and jitter buffer) is required.

You can install the devel packages:

```
yum install speex-devel
```

Or build the source by yourself:

```
wget http://downloads.xiph.org/releases/speex/speex-1.2beta3.tar.gz
tar -xvzf speex-1.2beta3.tar.gz
cd speex-1.2beta3
./configure --disable-oggtest --without-libogg && make && make install
```

4. Building **YASM**

YASM is only required if you want to enable VPX (VP8 video codec) or x264 (H.264 codec).

```
wget http://www.tortall.net/projects/yasm/releases/yasm-1.2.0.tar.gz
```

```
tar -xvzf yasm-1.2.0.tar.gz
cd yasm-1.2.0
./configure && make && make install
```

5. Building **libvpx**

Date: December 1, 2012.

libvpx adds support for VP8 and is optional but highly recommended if you want support for video when using Google Chrome or Mozilla Firefox.

You can install the devel packages:

```
sudo yum install libvpx-devel
```

Or build the source by yourself:

```
git clone http://git.chromium.org/webm/libvpx.git
cd libvpx
./configure --enable-realtime-only --enable-error-concealment --disable-examples --enable-vp8 --enable-pic --enable-shared --as=yasm
make && make install
```

6. Building **libyuv**

libyuv is optional. Adds support for video scaling and chroma conversion.

```
mkdir libyuv && cd libyuv
svn co http://src.chromium.org/svn/trunk/tools/depot_tools .
./gclient config http://libyuv.googlecode.com/svn/trunk
./gclient sync && cd trunk
make -j6 V=1 -r libyuv BUILDTYPE=Release
make -j6 V=1 -r libjpeg BUILDTYPE=Release
cp out/Release/obj.target/libyuv.a /usr/local/lib
cp out/Release/obj.target/third_party/libjpeg_turbo/libjpeg_turbo.a /usr/local/lib
mkdir --parents /usr/local/include/libyuv/libyuv
cp -rf include/libyuv.h /usr/local/include/libyuv
cp -rf include/libyuv/*.h /usr/local/include/libyuv/libyuv
```

7. Building **opencore-amr**

opencore-amr is optional. Adds support for AMR audio codec.

```
git clone git://opencore-amr.git.sourceforge.net/gitroot/opencore-amr/opencore-amr
autoreconf --install && ./configure && make && make install
```

8. Build **libopus**

libopus is optional but highly recommended as it's an MTI codec for WebRTC. Adds support for Opus audio codec.

```
wget http://downloads.xiph.org/releases/opus/opus-1.0.2.tar.gz
tar -xvzf opus-1.0.2.tar.gz
cd opus-1.0.2
```

```
./configure --with-pic --enable-float-approx && make && make install
```

9. Building **libgsm**

libgsm is optional. Adds support for GSM audio codec.

You can install the devel packages (**recommended**):

```
sudo yum install gsm-devel
```

Or build the source by yourself:

```
wget http://www.quut.com/gsm/gsm-1.0.13.tar.gz
tar -xvzf gsm-1.0.13.tar.gz
cd gsm-1.0-pl13 && make && make install
#cp -rf ./inc/* /usr/local/include
#cp -rf ./lib/* /usr/local/lib
```

10. Building **g729**

G729 is optional. Adds support for G.729 audio codec.

```
svn co http://g729.googlecode.com/svn/trunk/ g729b
cd g729b
./autogen.sh && ./configure --enable-static --disable-shared && make && make install
```

11. Building **iLBC**

iLBC is optional. Adds support for iLBC audio codec.

```
svn co
http://doubango.googlecode.com/svn/branches/2.0/doubango/thirdparties/scripts/ilbc
cd ilbc
wget http://www.ietf.org/rfc/rfc3951.txt
awk -f extract.awk rfc3951.txt
./autogen.sh && ./configure
make && make install
```

12. Building **x264**

Date: December 2, 2012

x264 is optional and adds support for H.264 video codec (requires FFmpeg).

```
wget ftp://ftp.videolan.org/pub/x264/snapshots/last_x264.tar.bz2
tar -xvjf last_x264.tar.bz2
# the output directory may be difference depending on the version and date
cd x264-snapshot-20121201-2245
./configure --enable-shared --enable-pic && make && make install
```

13. Building **FFmpeg**

Date: December 2, 2012

FFmpeg is optional and adds support for H.263, H.264 (requires x264) and MP4V-ES video codecs.

```
git clone git://source.ffmpeg.org/ffmpeg.git ffmpeg
cd ffmpeg

# grap a release branch
git checkout n1.2

# configure source code
./configure \
--extra-cflags="-fPIC" \
--extra-ldflags="-lpthread" \
\
--enable-pic --enable-memalign-hack --enable-pthreads \
--enable-shared --disable-static \
--disable-network --enable-pthreads \
--disable-ffmpeg --disable-ffplay --disable-ffserver --disable-ffprobe \
\
--enable-gpl \
\
--disable-debug

make && make install
```

14. Building **Doubango**

Minimal build

```
cd doubango && ./autogen.sh && ./configure --with-ssl --with-srtp --with-speexdsp
make && make install
```

Recommended build

```
cd doubango && ./autogen.sh && ./configure --with-ssl --with-srtp --with-speexdsp
--with-ffmpeg
make && make install
```

Full build

```
cd doubango && ./autogen.sh && ./configure --with-ssl --with-srtp --with-vpx --with-yuv
--with-amr --with-speex --with-speexdsp --with-gsm --with-ilbc --with-g729 --with-ffm-
peg
make && make install
```

5.2 Building webrtc2sip

webrtc2sip depends on Doubango IMS Framework v2.0 and libxml2.

The first step is to checkout the source code:

```
svn co http://webrtc2sip.googlecode.com/svn/trunk/ webrtc2sip
```

3. Installing libxml2

```
yum install libxml2-devel
```

4. Building webrtc2sip

```
export PREFIX=/opt/webrtc2sip
cd webrtc2sip && ./autogen.sh && ./configure --prefix=$PREFIX
make clean && make && make install
cp -f ./config.xml $PREFIX/sbin/config.xml
```

5.3 Running webrtc2sip

Running webrtc2sip is as easy as executing "webrtc2sip" binary file. Please note that it requires a valid configuration file. The default configuration file should be named "config.xml" and placed in the same folder as "webrtc2sip".

5.3.1 Command line arguments

	Available since	Description	Example
`--config=PATH`	2.1.0	Overrides the default path to the "config.xml" file.	`--config=/tmp/config.xml`
`--help`	2.1.0	Displays the help message	
`--version`	2.1.0	Displays the gateway version	

For more information on supported command line arguments, please execute ***`webrtc2sip --help`***.

6 Testing the gateway

Let's say the webrtc2sip gateway and SIP server are running on two different PCs with IP addresses equal to *192.168.0.1* and *192.168.0.2* respectively.

1. Open http://sipml5.org/expert.htm in your browser
2. Fill *"WebSocket Server URL"* field with the IP address and port where your webrtc2sip gateway is listening for incoming Websocket connections (e.g *ws://192.168.0.1:10060* or *wss://192.168.0.1:10062*). **IMPORTANT:** Do not forget the url scheme (**ws://** or **wss://**).
3. The *"SIP outbound Proxy URL"* is used to set the destination IP address and Port to use for all outgoing requests regardless the domain name (a.k.a **realm**). This is a good option for developers using a SIP domain name without valid DNS A/NAPTR/SRV records. E.g. *udp://192.168.0.2:5060.*
4. Check *"Enable RTCWeb Breaker"* if you want to call a SIP-legacy endpoint.

7 Interoperability

This section contains good tips to help you to debug some issues you can find when you're trying to make/receive calls to/from well-known SIP clients or servers using a web browser. Please note that if your preferred web browser is Google Chrome then, **we highly recommend using the STABLE version**.

7.1 Servers

This section explains know issues and how to tackle them.

7.1.1 Asterisk

Date: November 29, 2012

There are some issues (on both Asterisk and Chrome) to get both way audio and video when using Google Chrome stable. There are two solutions.

1. Patching Asterisk: This is only recommended if you're a developer and trying to learn new cool features. Please note that this will not allow video to flow as Asterisk doesn't support *VP8*. For more information on how to patch Asterisk, visit http://code.google.com/p/sipml5/wiki/Asterisk
2. Enabling the RTCWeb Breaker: This is the recommended solution and it allows both audio and video to flow. Video stream will flow even if the web browser and the SIP client/server do not share the same codecs (thanks to the *Media Coder* module).

7.1.2 FreeSWITCH

The problem here is that *FreeSWITCH* do not support *ICE* and some other mandatory *RTCWeb* features. Enabling the *RTCWeb Breaker* module (web browser side) is enough to fix the issue.

7.2 Web Browsers

7.2.1 Google Chrome

Date: November 29, 2012

We highly recommend using the STABLE version for your tests. Please note that we don't provide any kind of help or support if you're using the DEV or CANARY versions.

5. Chrome uses **SAVPF** profile. The **S** is for secure (SRTP) and the **F** for feedbacks (RFC 4585). If one of these features is not supported by the remote SIP client/server then you have to enable the *RTCWeb Breaker* module (web browser side).
6. Chrome only includes *VP8* video codec which is not supported by most of SIP clients/servers (e.g. xlite, Asterisk…). If your SIP client/server supports *H.264*, *H.263*, *Theora* or *MP4V-ES* then, you have to enable both the *RTCWeb Breaker* (web browser side) and *Media Coder* (server side) modules to have video. Please note that the *Media Coder* module will most likely not be enabled on the sipml5.org hosted servers.

7.2.2 Firefox Nightly

Date January 14, 2012

Right now only Nightly version of Firefox natively supports RTCWeb. The latest version known to work is **21.0a1 (2013-01-12)**. Please also note that there is a known issue on DTLS-SRTCP decoding (check issue 194 for more information).
The RTCWeb implementation in Firefox Nightly uses **DTLS-SRTP** while Chrome uses **SDES-SRTP** which means you need to enable the *RTCWeb Breaker* module to make calls from one browser to another.

7.2.3 Firefox, Safari, IE and Opera

Date: November 29, 2012

--This section intentionally left blank--

7.2.4 Ericsson Bowser

Date: November 29, 2012

Ericsson Bowser does not support Secure RTP (*SRTP*) and only include *H.264* video codec. Bowser can talk to most of SIP clients but is not compatible with Canary or any RTCWeb client.
Enabling the *RTCWeb Breaker* (browser side) will allow Bowser to talk to Chrome for audio only as G.711 is a common codec but video requires the *Media Coder* to be enabled (server side).

7.3 JavaScript SIP stacks

Date: November 29, 2012

--This section intentionally left blank--

8 Security issues

When the *RTCWeb Breaker* module is enabled on the client side (web browser) then, the server will act as a *b2bua* for all incoming and outgoing *INVITE*s to this web browser. Please note that this only apply to the SIP account tied to this particular web browser. Acting as a *b2bua* means the server will generate a completely new request for each *INVITE*. The new *INVITE* request from the *b2bua* could be challenged (*SIP 401/407 response*) by the remote SIP-legacy network which means the *b2bua* must have the SIP account credentials. Instead of sending the username and password to the *b2bua* we transmit an authentication token (***HA1***). Off course there is no possibility to retrieve the password from the token but it's highly recommended not to allow any intermediate node to intercept it and this is why sipML5 automatically use ***secure websocket (WSS)*** when *RTCWeb Breaker* is enabled.

```
HA1 = MD5(username:realm:password)
```

```
INVITE sip:1061@sip2sip.info SIP/2.0
Via: SIP/2.0/WSS df7jal23ls0d.invalid;branch=z9hG4bK1tvqE4UJ9VNwxbRNKODUvXQeoDUPL
w2W;rport
From: <sip:13131313@sip2sip.info>;tag=JA2uxtI28xUAM4ZyForT
To: <sip:1061@sip2sip.info>
Contact: "13131313"<sip:13131313@df7jal23ls0d.invalid;rtcweb-breaker=yes;transpo
rt=wss>;impi=13131313;ha1=050a0170e77b5d345388598f70d2d1bf;+sip.ice
Call-ID: e7c9abfc-67ce-3192-75e6-4429cbdf2626
CSeq: 9517 INVITE
```

The above *INVITE* request is received from the web browser when *RTCWeb Breaker* module is enabled. The *b2bua* will not include the ***HA1*** parameter when making a new INVITE to the SIP-legacy network even if a secure transport (e.g. *DTLS or TLS*) is used to forward it.